Mag./Giu./Luglio 1990, Anno V n. 3

La rivista per utenti di C-64/128 ed Amiga

COMMODORE GAZETTE

SPECIALE AMIGA 3000:

- TUTTO SUL NUOVO A3000
- L'ECS E IL VDE
- IL WORKBENCH 2.0
- A500/1000/2000: COSA FARE
- AMIGAVISION E LA MULTIMEDIALITÀ

Saloni USA:

- AMIEXPO
- WORLD OF AMIGA

Commodore 64/128:

LISTATI E PROGRAMMI MUSICALI

Programmare l'Amiga:

- IMPARARE IL C
- LE FUNZIONI ARP/DOS

Telecomunicazioni:

XMODEM

Novità per l'Amiga:

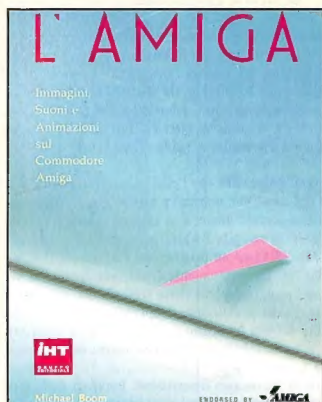
- DIGIVIEW 4.0
- SCHEDE 68030

WORKBENCH • AMIGADOS • EDITORS • UTILITIES

Using The System Software

Commodore AMIGA

Sped. in abb. post. Gr. III/70 – Lire 8.000

Un computer chiamato AMIGA

L'AMIGA

Come il Commodore Amiga ha cambiato il mondo dei computer, così il volume *L'Amiga* proietterà la vostra immaginazione lungo nuovi orizzonti di creatività. Preparatevi a ottenere stupefacenti risultati dal vostro computer: sofisticate immagini video, suoni e musica, sequenze animate da registrare su videocassetta, e molto di più. In questo volume troverete inoltre consigli utili per ottenere il meglio dall'Amiga Basic e da prodotti software eccezionali come Deluxe Paint, Deluxe Music e Deluxe Video.

416 pagine, oltre 100 illustrazioni, L. 60.000

IL MANUALE DELL'AMIGADOS

Questo volume è l'unica documentazione ufficiale realizzata dalla Commodore sul sistema operativo dei computer Amiga. Contiene tre libri (Il Manuale per l'utente, Il Manuale per il programmatore e Il Manuale di riferimento tecnico) che costituiscono nel loro complesso la guida più completa per ogni utente dell'Amiga, dal principiante che vuole usare i comandi dell'AmigaDOS (che non sono spiegati dall'opuscolo in dotazione al computer), fino al programmatore evoluto che troverà utili informazioni per programmare in C e in Assembly.

376 pagine, L. 60.000

I libri IHT sono disponibili nelle migliori librerie e computer shop. Per ordini diretti servirsi dell'apposito modulo pubblicato a pagina 127
IHT Gruppo Editoriale - Via Monte Napoleone, 9 - 20121 Milano - Tel. 02/794181-794122 - Fax 784021 - Telex 334261 IHT I
Distribuzione: RCS Rizzoli Libri - Via Scarsellini, 17 - 20161 Milano - Tel. 02/64068508

Sommario

FOTO PATRICIA LEEDS

MAG./GIU./LUGLIO 1990
Anno V, N.3

ARTICOLI

RUBRICHE

NOTE EDITORIALI

è una pubblicazione

Direttore responsabile: Massimiliano M. Lisa
Redazione: Mauro Gaffo, Dario Tonani, Nicolò Fontana Rava
Collaborazione editoriale: Fabio Rossi, Filippo Moretti, Dario Zanon, Avelino De Sabbata, Fulvio Piccioli, Alfredo Prochet, Michael Moosleitner, Stefano Franzato
Corrispondenti USA: William S. Freilich, Daniela D. Freilich
Collaborazione editoriale USA: Matthew Leeds, Ervin Bobo, Eugene P. Mortimore, Morton A. Kevelson. *AmigaWorld*: Bill Catchings, Mark L. Van Name, John Foust, Joel Hagen, Sheldon Leemon. *Run*: Leonard Morris, Thomas Czarnecki, John Ryan
Segretaria di redazione: Giovanna Varia
Impaginazione e grafica: Andrea De Michelis
Fotografie: A.&D., Patricia Leeds
Disegni: Marco Piazza, Giuseppe Festino

Direzione, Redazione, Amministrazione: IHT Gruppo Editoriale S.r.l. - Via Monte Napoleone, 9 - 20121 Milano
Fotocomposizione: IHT Gruppo Editoriale S.r.l. - Divisione grafica
Fotolito: Colour Separation Trust S.r.l. - Via Melchiorre Gioia, 61 - 20124 Milano
Stampa: OFSA S.p.A. - Via Giacomo Puccini, 64 - 20080 Casarile (MI)
Distribuzione per l'Italia: Messaggerie Periodici S.p.A. - V.le Famagosta, 75 - 20142 Milano - Tel. 02/8467545
Distribuzione per l'estero: A.I.E. S.p.A. - Via Gadames, 89 - 20151 Milano - Tel. 02/3012200

Pubblicità: IHT Gruppo Editoriale S.r.l. - Via Monte Napoleone, 9 - 20121 Milano - Tel. 02/794181-799492-792612-794122 - Fax 02/784021 - Telex 334261 IHT I - Agente pubblicitario: Aldo Pagano Pagano

Abbonamenti: IHT Gruppo Editoriale Servizio Abbonati - Via Monte Napoleone, 9 - 20121 Milano. Linea per registrazione abbonamenti: 02/794181 799492-792612-794122
Costo abbonamenti: Italia 12 numeri L. 96.000 - 24 numeri L. 192.000 - 36 numeri L. 288.000
Estero: Europa L. 150.000 (10 numeri) Americhe, Asia... L. 200.000 (10 numeri). I versamenti devono essere indirizzati a: IHT Gruppo Editoriale S.r.l. - Via Monte Napoleone, 9 - 20121 Milano mediante emissione di assegno bancario o vaglia postale

Arretrati: Ogni numero arretrato: L. 16.000 (sped. compresa)

Autorizzazione alla pubblicazione: Tribunale di Milano n. 623 del 21/12/85. Periodico mensile. Sped. in abb. post. gr. III/70. ISSN: 0394-6991
La IHT Gruppo Editoriale è iscritta nel Registro Nazionale della Stampa al n. 2148 vol. 22 foglio 377 in data 5/6/1987

Associato alla U.S.P.I. (Unione Stampa Periodica Italiana)

È arrivato!

Come vi avevamo preannunciato nel precedente numero di *Commodore Gazette*, l'Amiga 3000 è finalmente nato. E ci troviamo dinanzi al completamento della ricalibrazione della strategia Commodore.

La storia ha inizio nel 1985 con l'uscita dell'Amiga 1000. Doveva essere una macchina rivoluzionaria con un target ad ampio respiro: dall'hobbysta al professionista. Ben presto arrivò però un segnale chiaro da parte del mercato. Il messaggio diceva che la tecnologia Amiga aveva alcune carte in regola per un certo tipo di utenza professionale, ma che mancava un elemento importante. Si trattava della compatibilità MS-DOS, che peraltro sin dall'uscita della macchina si sosteneva ottenibile via software. La strada dell'emulatore software non era però destinata ad avere successo, e infatti il *Transformer* non raggiunse qualità di velocità e compatibilità tali da rendere l'Amiga in grado di disporre di un lato MS-DOS compatibile.

Fu così che venne realizzato il Sidecar, un'espansione hardware esterna piuttosto voluminosa, inelegante e costosa, che conteneva un vero e proprio clone XT da collegarsi allo slot di espansione dell'A1000.

Alla Commodore però, chi doveva decidere non se ne stava con le mani in mano. Divenne chiaro che era necessaria una macchina per l'hobbysta evoluto e il professionista, che offrisse caratteristiche di espandibilità interne. Allo stesso modo si delineò il fatto che per il mercato hobbystico era necessaria una macchina più snella, meno costosa e con un look più *consumer*. Fu così che nacquero l'A2000 e l'A500, computer che sostituirono l'A1000 e ne decretarono la scomparsa.

La tecnologia è però in continua evoluzione ed è arrivato il giorno in cui anche l'architettura dell'A2000 non è più così attuale. Oggi è necessaria un'architettura a 32 bit con un microprocessore del calibro del 68030. Inoltre si va verso la miniaturizzazione e il grosso cabinet dell'A2000 non è quanto di più attraente sia disponibile sul mercato. Ecco così che queste esigenze, unite a quelle di nuovi modi video e di alta risoluzione senza sfarfallio, hanno dato origine all'Amiga 3000. Una macchina tecnologicamente all'avanguardia, dal prezzo discretamente competitivo, che ha come obiettivo il mercato professionale.

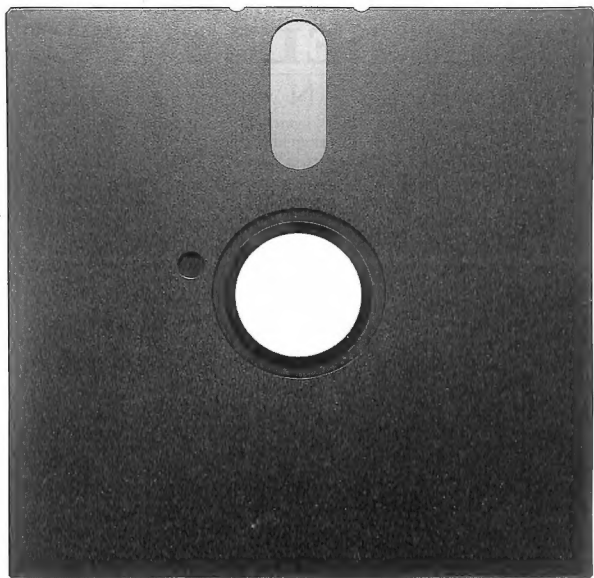
Un aspetto positivo della politica Commodore è che – come conferma anche Werter Mambelli (amministratore delegato della filiale italiana) – l'Amiga 3000 non vuole essere la macchina professionale per applicazioni quali database o contabilità, nelle quali il predominio MS-DOS è indiscusso. Si è deciso di cavalcare i cavalli di battaglia più consoni alle capacità del prodotto, proponendo l'Amiga 3000 come macchina per il professionista-creativo. E dal momento che oggi si parla tanto di multimedialità, si è dotato l'elaboratore di un sistema multimediale decisamente interessante, battezzato *AmigaVision*.

Per questa seconda metà degli anni Novanta la Commodore sembra quindi aver completato il "riaggiustamento del tiro" iniziato dopo l'uscita dell'A1000, proponendo l'A500 per l'hobbysta, l'A2000 per l'hobbysta evoluto e l'A3000 per il professionista.

Ma in realtà non è ancora finita. Il grande successo a livello mondiale delle *console* per videogiochi non è passato inosservato in casa Commodore... È infatti di questi giorni l'annuncio di un nuovo prodotto basato sulla tecnologia dell'Amiga 500.

Novità su Commodore Gazette

Con questo numero abbiamo introdotto una novità di un certo rilievo. Il corpo del carattere col quale viene impaginato il nostro periodico è stato ridotto. Le nuove dimensioni, oltre ad allinearsi con quanto propongono le altre riviste del settore, permettono di utilizzare più spazio. Il che vuol dire che su ogni singola pagina sono contenute molte più informazioni che in passato. Ma non è finita. Abbiamo deciso anche di offrire su ogni numero una maggiore varietà di articoli. W *Commodore Gazette*.

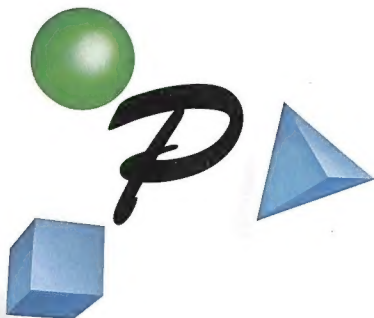
M.L.

DOMANDA: È COMPATIBILE?
RISPOSTA: CHE DOMANDE.

Da oggi, grazie ai PC Commodore, problemi tecnici e problemi pratici si risolvono più facilmente. Commodore Italiana, infatti, ha creato e garantisce in prima persona una linea di personal capace di rispondere alle esigenze di tutti e di lavorare e dialogare con tutti: dall'utente più sofisticato al neofita più acerbo. Da oggi, invece di scegliere un semplice PC, scegliete di fare un investimento garantito da Commodore Italiana.

Commodore

PC COMMODORE. FACILE IL DIFFICILE.

LA POSTA DELLA GAZETTE

LA VOCE DEI NOSTRI LETTORI

FAST-CATALOGER

Nel mondo dei computer da sempre esiste il problema di organizzare in modo semplice e funzionale la propria biblioteca di programmi, nella quale la confusione cresce in modo esponenziale, a mano a mano che viene aggiunto nuovo software. Fino ad ora, pur essendo vasta la schiera di utility per C-64 espressamente dedicate a questo scopo, non esisteva uno strumento che si potesse definire veramente efficiente e professionale. Per gli utenti di Commodore 128 da oggi è invece disponibile *Fast-Cataloger*. *Fast-Cataloger* risolve il problema della creazione e gestione di uno o più archivi del software posseduto (sia su disco che su cassetta), con una notevole flessibilità d'uso.

Questo sistema si avvale di un'interfaccia utente molto efficiente, organizzata in tre menu nei quali sono raccolte le opzioni riguardanti la gestione degli archivi, le operazioni di uscita dei dati, e le operazioni di gestione dei drive. In quest'ultimo menu è presente un set esteso di opzioni tra le quali troviamo quelle dedicate alle normali operazioni sui dischi (visualizzazione directory, formattazione dischi, cancellazione e rinominazione file...); inoltre annovera operazioni inusuali quali la modifica del nome del disco, il controllo che l'ID visualizzata corrisponda a quella di formattazione... Un apposito sottomenu dà la possibilità all'utente di personalizzare la configurazione di default per poterla ritrovare ad ogni utilizzo del programma. Tramite il menu di configurazione è possibile modificare le codizioni impostate per quanto riguarda le unità a disco, il monitor...

La semplicità d'uso del sistema è determinata in parte anche dal fatto che *Fast-Cataloger* è dotato di un HELP in linea intelligente: la pressione dell'omonimo tasto infatti richiama una finestra di aiuto con le informazioni in italiano relative all'operazione che si sta effettuando. Prerogativa peculiare del sistema è la possibilità di dividere, durante la catalogazione, i dati di un archivio in sei categorie diverse, ridefinibili per ogni archivio, grazie alle quali è più agevole risalire ai singoli file durante le ricerche.

Fast-Cataloger permette l'ingresso dei dati in modo automatico oppure in modo manuale. Nel modo automatico la directory del disco da catalogare (formattato in modo 1541 o in modo 1571) viene scandita dal sistema e i file incontrati vengono memorizzati in un buffer. I titoli incontrati, possono a questo punto essere catalogati in una delle categorie predisposte, oppure possono essere sottoposti a una fase di edit, durante la quale per ognuno viene definita la categoria di catalogazione, oppure l'esclusione della catalogazione. Durante questa fase sono possibili anche operazioni di modifica sui dati. Questa funzione consente di rendere catalogabili anche quei dischi con directory elaborate (ad esempio contenenti caratteri di controllo). Questi caratteri, non rappresentabili sullo schermo, vengono modificati da *Fast-Cataloger* nel carattere "?".

Nel modo manuale invece, i dati vengono inseriti da tastiera: è possibile inserire dati di nuove catalogazioni, oppure aggiungere dati a catalogazioni già esistenti. In questa modalità è inoltre possibile inserire i dati di file memorizzati su cassetta.

Fast-Cataloger non usa la memoria centrale per memorizzare i dati, i quali ad ogni catalogazione vengono registrati sul disco dedicato all'archivio. Questo rende possibile la mani-

Indirizzate tutta la corrispondenza per la rivista a:

COMMODORE GAZETTE
La posta della Gazette
Via Monte Napoleone, 9
20121 Milano

Preghiamo i lettori di essere concisi e concreti, per darci modo di rispondere al più grande numero possibile di lettere. La redazione si riserva comunque il diritto di sintetizzare le lettere troppo lunghe.

polazione di archivi di considerevoli dimensioni, con unica limitazione la capacità del disco dati. I dati di ogni archivio possono essere sottoposti alle operazioni di riordino e di ricerca sotto varie chiavi. In particolare le ricerche si avvalgono di un sistema che permette di estrarre dall'archivio tutti gli elementi che rispondono a determinati requisiti. È possibile sapere ad esempio quanti e quali siano i dischetti che abbiano un certo numero di blocchi liberi, oppure il nome dei file al cui interno sia contenuta una determinata sequenza di caratteri...

Oltre alla possibilità di produrre elenchi su carta in svariati formati, *Fast-Cataloger* presenta la possibilità di stampare le etichette delle catalogazioni effettuate, che possono essere applicate ai dischi o alle cassette catalogate.

Fast-Cataloger, accompagnato da un esauriente manuale utente, viene distribuito direttamente dall'autore al prezzo di lire 30.000 più le spese di spedizione. Eventuali ordini o ulteriori informazioni possono essere indirizzati a:

Avelino De Sabbata
Via G. Carducci, 3
33050 Pozzuolo del Friuli (UD)
(Tel. 0432-560426)

L'AMIGA ALL'UNIVERSITÀ

Sono un ricercatore del Dipartimento di Geografia dell'Università degli Studi di Bologna, mi occupo di studi di carattere ambientale utilizzando particolarmente immagini aeree e da satellite. L'elaborazione delle immagini riveste quindi un significato importante per le mie ricerche, e a tal fine mi sono interessato alle caratteristiche di elaborazione d'immagini proprie dei computer della serie Amiga.

Vi invio copia di una pubblicazione presentata al XXIV Convegno Nazionale dell'Associazione Italiana di Cartografia tenutosi a L'Aquila nel maggio 1988. In detto lavoro sono presenti alcuni esempi di applicazione di tecniche di image processing finalizzate allo studio di caratteri geo-ambientali. Gli strumenti utilizzati sono un Amiga 1000 dotato di espansione di memoria di almeno 1 Mega, un videodigitalizzatore, la Polaroid Palette e vari pacchetti per l'elaborazione d'immagini, quali *Pixmate*, *Butcher*, *Analitic Art*, *Deluxe Paint*. I risultati ottenuti, considerando il rapporto costo/prestazioni, sono veramente eccellenti e non sfigurano affatto nei confronti di quelli ottenuti con attrezzature dai costi enormemente superiori.

Il Commodore Amiga, purtroppo, appare ai più come un eccellente videogioco, ma quando si esce da questo ambito e si tenta, come più volte ho provato, di elogiarne anche le caratteristiche professionali, ci si sente quasi derisi in special modo dai possessori dei *famosi* Macintosh.

Personalmente ritengo superiore o almeno pari per prestazioni l'Amiga; devo però ammettere che da parte della casa madre Commodore non si fa molto per mutare questa immagine ludica del nostro computer; io stesso ho dovuto produrmi in proprio programmi di analisi delle superfici topografiche che il mercato non propone, per cui basterebbe promuovere la realizzazione di pacchetti applicativi scientifici di cui peraltro si avverte un notevole vuoto, almeno nel campo di mio prevalente interesse, cioè la topografia, geografia, geologia e cartografia, e, date le elevatissime capacità grafiche, si farebbero impallidire tutti gli altri elaboratori della medesima fascia di prezzo, Macintosh compreso.

Dr. Gabriele Giorgi
Dip. Geografia
Università degli Studi di Bologna

La ringraziamo per il documento inviatoci e pubblichiamo volentieri la Sua lettera a titolo di testimonianza.

PROGRAMMARE VIDEOGAME CON L'AMIGA

Tralasciamo i soliti complimenti e passiamo subito ai fatti. Io ho un Amiga 500 V. 1.2 e vorrei utilizzarlo per programmare giochi, visto che questo è il mio sogno nella vita. Io vorrei da voi un consiglio e cioè se comprare la *Zorro Big Blue Bus* con la scheda 68030 oppure aspettare la 68040. Soprattutto vorrei sapere da voi se è possibile collegare una di queste due schede nello slot della *Z.B.B.B*. Ho letto le recensioni delle due schede, ma come sempre accade nel mondo dei computer, la 68030 viene definita già "vecchia" visto che la 68040 ha la possibilità di trasferire 200 Mbyte d'informazioni al secondo, ha 4 Mbyte di cache, ecc., almeno questo diceva la rivista *MC Microcomputer* nel numero di febbraio 1990. Il costo sarebbe di $795, quindi un prezzo davvero basso.

A.B.
Località non spec.

Il consiglio che ci sentiamo di darLe è che per programmare non è assolutamente necessario espandere il computer con nuovi microprocessori, anche perché i possibili utenti del Suo videogioco non ne disporranno di certo. Pensi invece a fare esperienze di programmazione sulla Sua macchina. Se poi desidera spendere dei soldi...

UN COMMODORE POINT UN PO' SPECIALE

Tempo addietro avete gentilmente pubblicato una mia protesta riguardante i bidoni di un Commodore Point. Mi sembra quindi corretto segnalarvi un Commodore Point gestito da persone efficienti e disponibili. Sto parlando della Logitek s.r.l. di Milano.

Invece delle solite informazioni, ci hanno fornito una serie di consulenze specifiche e gratuite su ciò che ci serviva; ed hanno continuato a seguirci anche dopo l'acquisto. Hanno installato per noi Amiga ed accessori. Il sig. Franco Francia è rimasto in negozio con noi fino a notte fonda per far funzionare una Palette Polaroid, e grazie a lui le diapositive sono arrivate ad una conferenza in tempo.

Spero che molti vostri lettori che hanno scoperto un buon concessionario di S/W o H/W vincano la pigrizia e si decidano a scriverne. È un servizio utile per chi deve spendere i propri soldi ed un giusto riconoscimento per i commercianti competenti e cortesi.

Francesco G. Fantauzzi
Milano

FLOPPERIA
®
SRL

SOFTWARE GALLERY

UNA GUIDA PER ORIENTARSI NEL MONDO DEL SOFTWARE

DIGI-PAINT 3

Buoni effetti e grande semplicità, ecco la ricetta della perfezione

Computer: Amiga
Supporto: Disco
Prezzo: L. 150.000
Produzione: NewTek
Disponibile presso: Lago (Via Napoleona 16, 22100 Como – 031/300174)

L'uso del personal/home computer in campo artistico è una pratica abbastanza recente: è stato soprattutto l'avvento dell'Amiga, con i suoi 4096 colori in alta risoluzione e a basso costo, che ne ha reso possibile la diffusione.

Servirsi di questo nuovo strumento, però, significa anche acquisire un nuovo modo di percepire e d'intendere l'opera artistica, e implica quindi uno sforzo notevole sia per chi esamina il frutto della nuova tecnica sia, soprattutto, per l'artista. Vi sono però tali e tante nuove possibilità a disposizione di chi si cimenta con il computer, che l'eccitazione intellettuale spazza via ben presto ogni difficoltà.

Una delle caratteristiche più interessanti di *Digi-Paint 3*, il programma di cui ci occupiamo in questa recensione, è per esempio la possibilità di creare, manipolare e visualizzare immagini in modo Hold And Modify (HAM), ovvero con 4096 colori contemporaneamente presenti sullo schermo. Il modo HAM, tra l'altro, rende possibile generare effetti speciali come le trasparenze.

Il programma dispone di undici diversi modi di disegno. Quello di default è la bassa risoluzione, normale o interlace, ma la risoluzione di schermo può arrivare fino a 1024 x 1024 pixel se si dispone di sufficiente memoria. Per esempio, con una chip RAM di 1 megabyte si può arrivare fino a 960 x 960 pixel di risoluzione.

Il programma legge e scrive file in standard IFF, permettendo così una completa compatibilità con i prodotti che utilizzano lo stesso standard. Sempre in tema di compatibilità, segnaliamo quella con lo standard *AREXX*, che rende il prodotto "pilotabile" dall'esterno tramite appositi file di comandi. È inoltre possibile creare pennelli con qualsiasi elemento grafico e utilizzarli in tutti gli undici modi grafici disponibili. Se necessario è possibile utilizzare in parallelo due schermi grafici indipendenti, cosa che permette di mescolare tra loro gli elementi delle due pagine.

Incluso nel pacchetto, e quindi nel prezzo, troviamo il programma *Transfer 24* che consente di alterare in molti modi le immagini e di convertirle tra i diversi formati grafici dell'Amiga, inclusa l'alta risoluzione overscan con 768 x 480 pixel. Uno dei punti di forza di *Digi-Paint 3* è la completa aderenza del programma alle convenzioni dell'Amiga sull'uso del mouse e dei menu. A questa semplicità d'uso contribuisce senza dubbio la simpatica interfaccia grafica, che oltre a essere esteticamente gradevole, consente un rapido accesso a moltissime funzioni del program-

SCHEDA CRITICA

INSUFFICIENTE
Un pessimo prodotto che non merita nessuna considerazione.

MEDIOCRE
Il programma ha alcuni difetti di fondo, anche se nel complesso raggiunge quasi la sufficienza.

DISCRETO
Un prodotto accettabile, ma non aspettatevi grandissime emozioni.

BUONO
Raccomandato vivamente: tra i migliori programmi della sua categoria.

OTTIMO
Eccezionale! Fino ad oggi non si era mai visto niente del genere.

ma senza la necessità di continue consultazioni del manuale. Tutte le opzioni non accessibili tramite icone vengono raggiunte tramite semplici menu a discesa oppure combinazioni di tasti. Questa diversificazione dei comandi colloca *Digi-Paint 3* tra i programmi più flessibili e semplici da usare che ci sia mai capitato di provare.

Anche se viene dichiarato compatibile con tutta la famiglia di computer Amiga, i migliori risultati si ottengono con due disk drive e almeno 1 megabyte di RAM. Con una scheda acceleratrice, inoltre, si ottiene un sensibile incremento di prestazioni dal momento che il programma è stato progettato per sfruttarne la presenza nel modo migliore.

Per tornare al paragone con la pittura "classica", in *Digi-Paint* lo schermo del monitor rappresenta la tela su cui si sovrappone la "scatola degli attrezzi" (toolbox). In realtà, i toolbox disponibili sono quattro.

Tutti i toolbox hanno in comune la barra dei colori sulla parte superiore dello schermo. Le 16 caselle di sinistra mostrano i 16 colori della palette, poi viene il colore in uso e quindi le tre caselle per le sfumature. Altri elementi comuni a tutti i toolbox sono il consueto gadget per chiudere la finestra e i due gadget per il cicling. Ma vediamo in dettaglio questi quattro pannelli di controllo.

Il toolbox strumenti. Mette a disposizione gli strumenti per il disegno di cerchi, ellissi, rettangoli e poligoni (pieni o vuoti), e per il disegno a mano libera. Il pennello di disegno ha sette dimensioni e sette forme possibili, tutte combinabili tra loro; oppure può essere definito ex-novo dall'utente. Uno strumento molto utile è lo Zoom, rappresentato da un'icona a forma di lente, che permette d'ingrandire otto volte una zona dell'immagine e di lavorarvi

utilizzando tutti gli strumenti di *Digi-Paint*. La finestra di zoom è una normale finestra, che può essere ridimensionata e spostata a piacimento. Molto versatili sono le Forbici, che permettono di "ritagliare" una parte dell'immagine e di usarla come pennello.

Ma i comandi più potenti e versatili di questo toolbox, e probabilmente dell'intero programma, sono Undo e Repeat. Il primo dei due permette di ripristinare la situazione esistente prima dell'ultimo intervento, mentre il secondo ordina al computer di ripetere l'ultima azione eseguita. Finché non si prova a usarle, non ci si rende conto di quanto siano utili queste opzioni. Per esempio permettono di mettere alla prova vari gradi di trasparenza, alla ricerca di quello che dà l'effetto migliore. Basta impostare i parametri e disegnare l'oggetto, dopodiché si seleziona Undo, si varia un parametro e si seleziona Repeat: l'oggetto verrà ridisegnato con le nuove caratteristiche. Ma la funzione Repeat si rivela uno strumento prezioso anche in altre occasioni. Una per tutte, la creazione di un pennello personalizzato di forma irregolare: si crea un profilo disegnando a mano libera, si seleziona l'icona Forbici e quindi Repeat. Il profilo precedentemente creato viene ritagliato e diventa un pennello

pronto all'uso. Semplice no?

Il toolbox testi. Sul versante dei testi, *Digi-Paint* si rivela apparentemente un po' scarno. Fornisce infatti soltanto 6 fonti-carattere: Cinnamon, Diamond, Melba, Ruby, Texas e Topaz. A parte Cinnamon e Melba, che offrono due corpi (ovvero due possibili grandezza per il carattere), le altre mettono a disposizione un unico corpo. Anche sul versante degli stili notiamo una certa povertà; sono infatti disponibili soltanto il neretto, il corsivo e il sottolineato. La facilità con cui si usano i testi, tuttavia, fa dimenticare ben presto queste limitazioni: dopo aver inserito il testo desiderato, questo viene trattato come un pennello personalizzato, pertanto è molto semplice spostarlo e modificarlo con i vari strumenti grafici. Come al solito è possibile provare svariate combinazioni cambiando soltanto alcuni parametri. Si possono ottenere testi sfumati, multicolori, semitrasparenti e deformati in mille

modi... insomma, una varietà tale di scelte che non si sente certo la mancanza di altre fonti-carattere.

Il toolbox palette. Selezionando l'icona che raffigura una tavolozza si entra nel terzo toolbox, che permette di agire sui colori di default. Come tutti gli altri, anche questo toolbox presenta nella parte superiore la barra dedicata ai colori: le 16 caselle di sinistra mostrano i 16 colori principali della tavolozza, e sotto a ogni casella vi è una colonna che mostra tutti i toni disponibili per quel colore. Per selezionare un tono basta portarvi sopra il puntatore e premere il pulsante sinistro del mouse. Sotto alla casella del colore in uso ci sono tre selettori che mostrano in che percentuale sono mescolati i tre colori fondamentali (rosso, verde, blu). Ogni selettore può assumere 16 diverse posizioni, così le combinazioni possibili risultano 4096 che è appunto il numero massimo di colori disponibili.

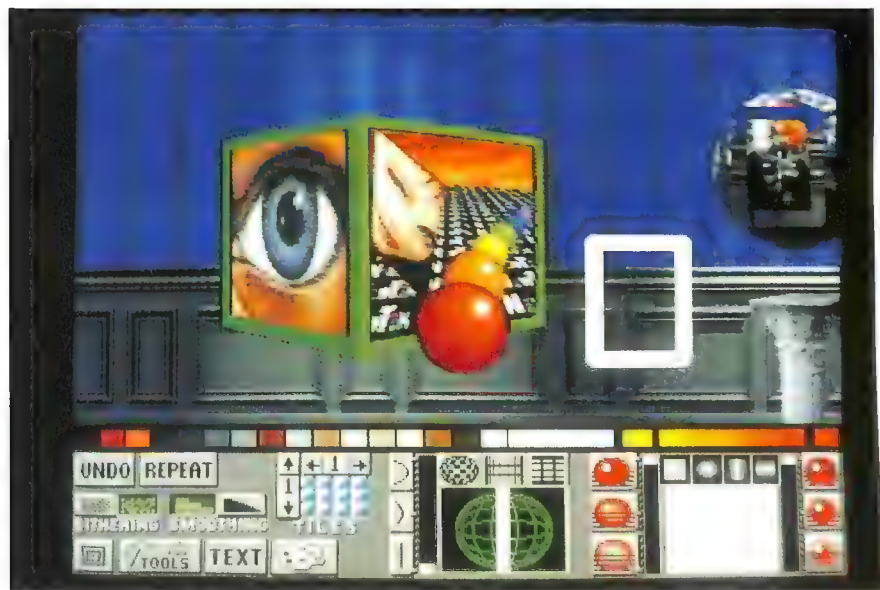
L'ultima opzione di questo toolbox è l'icona Pick, che permette di catturare un colore direttamente dallo schermo e di utilizzarlo come colore in uso o di modificarlo a piacimento.

Il toolbox controlli. Si tratta senza dubbio del "pannello di comando" più difficile da padroneggiare, ma è anche il più affascinante. Da qui possiamo infatti controllare gli effetti di trasparenza e di deformazione.

Gli effetti deformanti si attivano soltanto creando un pennello personalizzato e attivando il modo grafico TxMap. Dopodiché la deformazione, più o meno marcata, viene attivata semplicemente spostando un cursore che si trova sulla sinistra del rettangolo. Un elemento che influenza l'effetto finale è l'asse o il punto rispetto al quale viene deformato l'oggetto prescelto. Ci viene richiesto infatti di dichiarare, sempre tramite icone, se l'oggetto dev'essere deformato rispetto a un punto o rispetto a un asse (e in quest'ultimo caso dobbiamo specificare se l'asse è verticale o orizzontale). Subito sotto a queste icone è disegnata una sfera che rappresenta il nostro oggetto, e che costituisce il punto di riferimento rispetto al quale collocare l'asse o il punto.

Accanto ai comandi di deformazione c'è l'icona Tiles. Grazie a quest'ultima si può sdoppiare il pennello fino a nove volte in orizzontale o in verticale, generando curiosi effetti

ottici, molto utili per creare l'illusione della prospettiva.

Il toolbox controlli mette a disposizione inoltre la gestione delle trasparenze e delle sfumature di colore. I comandi che gestiscono queste funzioni sono collocati a destra di quelli delle deformazioni. L'interfaccia

utente è raffigurata anche qui da una sfera. Ai lati ci sono i controlli per la trasparenza (a sinistra per la trasparenza interna e a destra per i bordi). Per regolarli basta far scorrere un cursore lungo una scala grafica. Al di sopra della sfera ci sono quattro icone che sovrintendono alla regolazione delle sfumature. La prima a sinistra "spegne" l'uso delle sfumature, le altre tre permettono di selezionare il punto di applicazione delle sfumature con un criterio analogo a quello usato per determinare l'asse delle deformazioni.

Qui si conclude la panoramica sui toolbox. Alcune funzioni sono state descritte per sommi capi, e a quelle secondarie non abbiamo nemmeno accennato. Ma è doveroso riservare qualche parola anche ai menu. Come in tutti i programmi Amiga, i menu si attivano col pulsante destro del mouse e compiendo quest'operazione il toolbox in uso viene sostituito dalla finestra dei menu. In tutto ce ne sono sei. Il primo, **Picture**, permette di caricare e salvare immagini su disco, di pulire lo schermo e di determinarne le dimensioni. Grazie a questo menu si gestiscono i due schermi paralleli a cui abbiamo già accennato. Sempre attraverso Picture è possibile stampare il disegno finale. A questo proposito dobbiamo sottolineare che *Digi-Paint 3* non fa uso di driver dedicati ma utilizza quelli di Preferences.

Il secondo menu, **Brush**, permette di gestire i pennelli personalizzati, mentre il menu **Palette** consente ulteriori manipolazioni dei colori. Il menu **Effect** permette di ruotare e invertire le immagini.

Il menu **Mode** apre le porte degli undici modi grafici; vediamone alcuni. Range permette di utilizzare le sfumature e gli effetti di trasparenza. Colorize permette di cambiare i valori dei colori sullo schermo senza cambiarne la luminosità (è utile per colorare immagini in bianco e nero). Blur è in grado di sfumare i bordi nel passaggio da un colore all'altro. TxMap attiva i già descritti comandi di deformazione, ma può entrare in funzione solo a patto che sia stato creato un pennello personalizzato e che ne sia stata fatta una copia grazie alla voce "Copy this Brush". L'ultimo menu è **Prefs** che tra le altre cose permette di aprire o chiudere lo schermo *Workbench*.

È il momento delle conclusioni, anche se ci sarebbe ancora molto da dire su questo programma, che ha suscitato gli entusiasmi della redazione. Vi assicuriamo che è veramente semplice imparare a usarlo, anche perché il manuale in dotazione (in inglese) è molto chiaro e guida l'utente passo per passo nel mondo di *Digi-Paint 3*.

Il verdetto è più che positivo. Una nota di plauso, quindi, ai programmatori della NewTek che hanno svolto un ottimo lavoro realizzando un prodotto potente e al tempo stesso semplice da usare, che costituisce una base perfetta su cui esprimere la propria creatività. **F.P.**

OPERATION THUNDERBOLT

Un gioco che piacerebbe a Stallone e Schwarzenegger... insieme!

Computer: C-64/128/Amiga
Supporto: Cassetta/Disco
Prezzo: L. 18.000/25.000/29.000
Produzione: Ocean
Distribuzione: Leader (Via Mazzini 15, 21020 Casciago – 0332/212255)

Di *Operation Wolf* si è parlato molto l'anno scorso. Il giocatore impersonava un marine americano armato di mitraglietta Uzi con lanciagranate, impegnato in una missione di liberazione di ostaggi in Medio-riente. I bersagli erano, naturalmente, mezzi militari e guerriglieri disegnati in modo piuttosto realistico. Quella conversione da coin-op, ad opera della Ocean, vinse numerosi premi.

Operation Thunderbolt è il seguito di *Wolf*. L'ambientazione è simile, ma c'è una grossa differenza: questa volta le mitragliatrici, i lanciagranate e i giocatori sono due.

L'azione è vista in soggettiva, su scenari a scroll orizzontale o tridimensionali, nei quali i personaggi avanzano rapidamente con un bell'effetto simile a quello visto in *Space Harrier*. Il joystick controlla un mirino (inizialmente invisibile) che copre tutta l'area di schermo, mentre il pulsante funge ovviamente da grilletto. Lo scopo del gioco è di eliminare gli avversari prima che questi riescano a spararci addosso diminuendo la nostra energia vitale.

L'energia si può recuperare colpendo bersagli particolari, e la stessa cosa accade per le munizioni. Di quando in quando è possibile colpire e incamerare bonus vari, fra i quali un puntatore laser, un giubbotto antiproiettile e via dicendo.

Ancora una volta i bersagli sono guerriglieri nella tipica kefiah, carri armati, autoblindo, elicotteri e installazioni fisse. Per i più spietati c'è la possibilità di sparare a cani, gatti e civili che attraversano rapidamente lo schermo, spesso guadagnando munizioni extra. Nella versione per Amiga, fra una schermata e l'altra si possono ammirare i due eroi della vicenda minacciare i loro prigionieri per estorcere le informazioni necessarie a raggiungere il livello successivo. Sia sul C-64 che sull'Amiga la qualità tecnica del programma è notevolissima, con animazioni fluide, ottimi effetti tridimensionali e un buon sonoro. Un bellissimo programma. Un brutto gioco: la violenza non è il mio forte! **F.R.**

Le versioni Amiga (sopra) e C-64 (sotto)

FUTURE WARS

C'è qualcosa che non va nell'editor della Delphine...

Computer: Amiga
Supporto: Disco
Prezzo: L. 29.000
Produzione: Delphine Software
Distribuzione: Lago (Via Napoleona 16, 22100 Como – 031/300174)

Nella recensione di *Leisure Suit Larry III* (pagina 15) parliamo di editor per avventure grafiche. Tutte le software house sembrano ormai voler produrre giochi di questo tipo, ed è indispensabile avere un buon sistema di sviluppo per realizzare giochi divertenti e concorrenziali. La francese Delphine Software è stata la prima casa europea a sfidare le statunitensi Sierra e LucasFilm sul loro terreno, con la creazione del sistema impiegato per la realizzazione di questo *Future Wars*.

Ciò che salta subito all'occhio caricando il programma (e superando la noiosa procedura di identificazione-protezione) è la qualità della grafica: il personaggio controllato dal giocatore si muove su scenari a dir poco grandiosi, dettagliatissimi e ampi quanto l'intero schermo. Il protagonista è rappresentato da uno sprite piccino piccino, le cui dimensioni ridotte fanno ancor più risaltare i panorami nei quali si svolge l'avventura.

Come nei giochi della Sierra, il giocatore controlla direttamente i movimenti del personaggio, accedendo a una dettagliata interfaccia utente grazie alla semplice pressione di un pulsante. Da qui, sempre utilizzando solo il mouse, è possibile manipolare gli oggetti che si trovano nel corso dell'avventura e accedere ai comandi di sistema. Che si voglia raccogliere un sasso o salvare una partita, ogni volta il programma blocca l'azione e fa comparire sullo schermo una serie di finestre e sotto-finestre contenenti i menu necessari per compiere l'azione richiesta. Attivando l'ultimo comando della serie l'azione riprende, e si vede il protagonista che esegue l'ordine impartito.

La trama di *Future Wars* mette il giocatore nei panni di un pulitore di vetri ansioso di prendersi una rivincita contro il tirannico datore di lavoro. Ma quello che comincia come uno scherzo innocente, si conclude con una fuga precipitosa dagli uffici della megasocietà dove il personaggio lavora. Da qui in poi la vicenda prosegue su una linea fantascientifica, con teletrasportatori, macchine del tempo, razze aliene in guerra, catastrofi nucleari e mostri mutanti.

Purtroppo, la varietà della trama è anche uno dei grandi difetti del gioco. Per rendere coerente la storia, i programmatori sono stati costretti a inserire lunghi e forzatissimi dialoghi, intermezzi contenenti spiegazioni completamente illogiche e ogni sorta di fantasiose balordaggini. Nonostante gli sforzi, però, gli avvenimenti appaiono slegati e quasi casuali: l'impressione generale è di trovarsi sballottati da una parte all'altra dello spazio-tempo, costretti a fare cose di cui non si riesce a vedere lo scopo.

La struttura sequenziale dell'avventura costituisce un altro punto a sfavore. In realtà *qualunque* avventura grafica possiede uno schema più o meno lineare, ma *Future Wars* è forse l'unica in cui non sia possibile tornare sui propri passi. Ciò comporta un'alta probabilità di trovarsi in situazioni non risolvibili, ad esempio perché non si è preso nota di un'informazione o non si è raccolto un oggetto apparentemente inutile. Arrivare alle ultime battute del gioco e non poterlo concludere perché ci siamo dimenticati di prendere un oggetto nei primi schermi, e dover quindi ricominciare tutto daccapo è una delle cose più frustranti che possano capitare nel mondo dei videogiochi, e in *Future Wars* questa situazione sembra essere la norma.

L'ultimo appunto negativo va fatto all'interfaccia-utente, che è quanto di più ostico ci sia mai capitato sullo schermo. Benché l'idea delle finestre non sia malvagia, queste hanno la spiacevole tendenza a sovrapporsi l'un l'altra o ai fondali, risultando in definitiva totalmente illeggibili. Naturalmente non c'è modo di rileggere i messaggi "persi", e lottare con i menu è comunque fastidioso (e letale) nelle molte situazioni in cui ci si dovrebbe muovere in fretta per evitare un pericolo incombente. Per quanto riguarda la tanto decantata "interattività" dello schermo, poi, siamo al cospetto di una vera e propria tortura psicologica: il movimento del personaggio è clamorosamente impreciso, e tuttavia il gioco pretende a volte complicati slalom negli scenari contro strettissimi limiti di tempo, avvicinamenti millimetrici a pericoli mortali e cose simili.

Individuare gli oggetti sullo schermo a volte diventa una farsa. C'è una scena dove bisogna addirittura individuare un oggetto delle dimensioni di *un pixel*, che va poi raccolto da una posizione precisa e, ovviamente, del tutto arbitraria e illogica.

Detto questo, probabilmente non servono altri commenti. L'avventura è in italiano (più o meno), la grafica e il sonoro sono belli. Tutto il resto no.

F.R.

DRAGON'S LAIR: ESCAPE FROM SINGE'S CASTLE

Continua la saga di Dirk the Daring, la prima "laseravventura" per Amiga

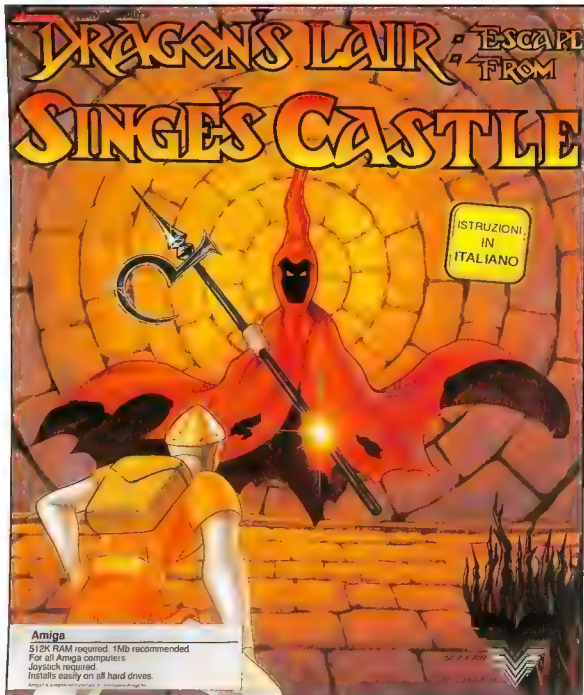
Computer: Amiga
Supporto: Disco
Prezzo: L. 99.000
Produzione: Bethesda Softworks
Distribuzione: Leader (Via Mazzini 15, 21020 Casciago – 0332/212255)

Il terribile Shapeshifter (mutaforma) vi ha sottratto un'altra volta la principessa Daphne, portandola nelle catacombe del castello dove vive il Dragone Singe.

Questa nuova avventura ha permesso ai programmatori del secondo *Dragon's Lair* d'introdurre altre scene tratte dal lasergame originale, e di crearne di nuove. Ritroviamo quindi alcune delle sequenze più belle del gioco originale (per esempio il cavallo meccanico o il re rana) che integrano e completano la narrazione delle traversie di Dirk. Tutto ciò, inoltre, è reso più interessante dalla possibilità di "fondere" insieme i due programmi (tramite un'apposita funzione "link") in modo da ottenere un'unica sequenza di scene.

Il nuovo programma, inoltre, mette a disposizione alcune scelte di configurazione che mancavano nella prima versione. Come sappiamo, in *Dragon's Lair* ogni scena veniva ripetuta una seconda volta "rovesciata" specularmente. In *Escape*, invece, un'opzione ci permette di disabilitare questa funzione e il giocatore può vedere ogni scena una volta sola. Però, attenzione: il verso di ogni sequenza è casuale, per cui non è detto che l'immagine vada sempre nella stessa direzione. Un'altra funzione fa comparire un riquadro nel quale viene indicata con un attimo d'anticipo la mossa da eseguire; questo aiuto diventa però sempre meno tempestivo a mano a mano che si avanza nell'avventura, fino a scomparire nelle scene finali. Se è stata eseguita la fusione dei due programmi, la funzione di aiuto è utilizzabile anche con il primo. Vi sono poi le funzioni già disponibili nella prima "puntata" del programma, che comprendono la possibilità di salvare la partita in corso, la scelta del numero di vite (da tre a cinque) e alcune routine per l'installazione dei due programmi su hard disk (ricordiamo che il trasferimento della prima parte era possibile soltanto con drive Comspec, mentre ora questa limitazione è stata superata).

Quanto alle considerazioni sul valore del programma, diciamo subito che non vi sono sostanziali miglioramenti né per quanto riguarda la grafica, né per quanto riguarda la giocabilità. Le scene sono comunque realizzate in maniera pressoché perfetta, come nel predecessore, e ricalcano fedelmente quelle del famoso lasergame. Come già accennato, la necessità di dotare *Escape* di un finale autonomo, ha fatto sì che venissero create anche alcune scene inedite. Purtroppo le nuove scene mettono tristemente in evidenza la mancanza d'ispirazione dei disegnatori e danno vita a un finale che lascia un po' di amaro in bocca.

In definitiva il discorso da fare è sempre lo stesso: questo tipo di giochi può piacere o meno. Chi ha apprezzato il primo corra subito ad acquistare una copia di *Escape*. Chi non ha mai visto il gioco stia attento: la splendida grafica annulla totalmente la normale interattività dei computer game. Chi invece trova questi giochi noiosi e inutili sa già cosa pensare di *Dragon's Lair: Escape from Singe's Castle*. Diplomaticamente vostro... **N.F.R.**

LEISURE SUIT LARRY III

Continua l'evoluzione delle avventure grafiche "made in Sierra"

Computer: Amiga
Supporto: Disco
Prezzo: L. 69.000
Produzione: Sierra
Distribuzione: Leader (Via Mazzini 15, 21020 Casciago – 0332/212255)

Attualmente esistono soltanto tre sistemi per la creazione di avventure grafiche: lo SCUMM della LucasFilm (*Zak McKraken & the Alien Mindbenders*, *Maniac Mansion*), quello senza nome della francese Delphine (*Future Wars*) e quello ormai collaudatissimo in possesso dei programmatori della Sierra OnLine. La più recente realizzazione del team Sierra riguarda il capitolo finale delle avventure del playboy Larry Laffer, meglio conosciuto come "Leisure Suit Larry".

Le due precedenti avventure del nostro eroe lo avevano portato a "mettere la testa a posto", sposando nientemeno che la ricca e bellissima principessa di un atollo polinesiano. Il gioco è ambientato proprio su quest'isola, famosa per le sue attrattive turistiche.

Già dalle prime battute le cose si mettono al peggio per Larry che, colto sul fatto dalla moglie mentre osserva con un cannocchiale le grazie di una turista, viene privato del suo status di principe consorte. Rimasto senza vestiti, senza denaro e soprattutto senza moglie, l'incorreggibile Larry comincia una nuova odissea per conquistare il cuore di una cantante di piano bar: Patti.

La prima parte del gioco è tutta rivolta ai tentativi di seduzione del playboy. Le situazioni comiche sono moltissime, e quasi perfettamente equilibrate da quelle "osé". A questo proposito vi anticipiamo che all'inizio del gioco viene proposto un test per scoprire l'età dell'utente attraverso domande di cultura generale (tipo "Quale delle seguenti canzoni non era suonata dai Beatles?"). A seconda dell'età stimata il gioco presenterà scene secondarie più o meno spinte, e al livello più alto (25 o più anni) sequenze come gli incontri di Larry con le sue partner diventano piuttosto esplicite, pur mantenendosi sempre nei limiti del buongusto.

Una volta riuscito a "conoscere" Patti (in senso biblico), la vicenda subisce una svolta. Larry comincia a svignarsela in maniera non troppo onorevole per sopravvivere agli attacchi della ragazza ormai scatenata, e il giocatore passa a controllare proprio quest'ultima. Grazie allo scambio di ruoli, l'utente vive gli stessi avvenimenti sotto una prospettiva nuova, attraversando così situazioni ancora più spassose di quelle sperimentate nei panni di Larry.

Rispetto ai precedenti prodotti di questa software house, la qualità generale del programma sembra essere migliorata non poco. Sul piano grafico abbiamo notato con piacere un più diffuso impiego delle animazioni, inframmezzate da ottime schermate statiche. Come sempre questo aspetto del programma risente del fatto che il gioco è stato concepito inizialmente per i PC, ma rimane tuttavia a livelli accettabili. Il sonoro – aspetto piuttosto trascurato nei precedenti titoli della serie – è anch'esso migliorato, grazie soprattutto al fatto che le musiche originali della versione PC sono state create pensando all'impiego di una scheda AdLib.

Come tutti i giochi prodotti dalla Sierra, anche *Leisure Suit Larry III* non è tuttavia esente da critiche, e la prima riguarda senz'altro il ritmo dell'azione. Fra spostamenti da un posto all'altro, caricamenti a ogni cambio di scena, intermezzi animati e "liti" con il sistema di controllo mouse/tastiera, in certi momenti il programma procede a velocità geriatrica. Il fatto di essere una semplice trasposizione della versione per PC crea non pochi limiti alla versione Amiga, costringendo il computer a operare a un decimo delle sue potenzialità. Il parser, poi, manifesta sempre più la sua inadeguatezza rispetto agli standard attuali. Nessuno pretende interfacce utente "alla Magnetic Scrolls" in giochi di questo genere, ma troviamo inaccettabile dover lottare con la limitatezza del vocabolario in pieno 1990, oltretutto utilizzando uno dei più potenti home computer in circolazione.

Detto questo, *Leisure Suit Larry III* rimane un'avventura grafica godibilissima per l'originalità della trama e per le numerose trovate umoristiche. Chi segue i giochi prodotti da questa casa apprezzerà le numerose citazioni sparse qua e là (non ultima la presenza degli stessi autori e del personale della Sierra in qualità di personaggi secondari).

Il nostro parere nei confronti di questo programma non può che essere favorevole, pur con tutte le imperfezioni elencate. L'avventura è abbastanza semplice da poter interessare anche i neofiti del genere, e non bisogna dimenticare l'esistenza degli "Hint Book", disponibili presso gli importatori, che contengono le soluzioni di tutti i giochi Sierra.

F.R.

688 ATTACK SUB

La "Sindrome Ottobre Rosso" comincia a colpire

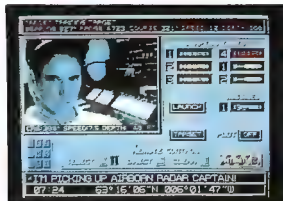
Computer: Amiga
Supporto: Disco
Prezzo: L. 45.000
Produzione: Electronic Arts
Distribuzione: C.T.O. (Via Piemonte 7/F, 40069 Zola Predosa – 051/753133)

Abbiamo già notato in passato quanto sia curiosa l'altalena delle tendenze nel campo delle simulazioni militari: c'è stato il periodo degli aerei "invisibili", degli elicotteri, dei carri armati... e adesso ecco arrivare l'era dei sommergibili. Non è difficile indovinarne la ragione: oltre a essere mezzi molto interessanti da manovrare, i sottomarini saranno anche i protagonisti di due pubblicizzatissimi film hollywoodiani (uno dei quali è *The Hunt for Red October*, con Sean Connery, tratto dal romanzo di Tom Clancy). L'industria dei videogiochi ha spesso beneficiato di simili fenomeni extramediali, e non si poteva certo lasciar scappare un'occasione simile.

688 Attack Sub è centrato sulle avventure fantapolitico-militari del sottomarino americano "688 – Los Angeles" e del suo antagonista sovietico "Alfa". Una delle caratteristiche più interessanti del programma è proprio la possibilità di affrontare missioni analoghe con i due mezzi, che si trovano così ad avere obiettivi diametralmente opposti. Le missioni vanno da azioni punitive contro le navi libiche a fughe "dimostrative" attraverso lo stretto di Gibilterra, dall'appoggio di una controrivoluzione in Jugoslavia a missioni nelle acque giapponesi durante la Terza guerra mondiale.

L'ultimo scenario culmina in un attacco globale contro l'invasore russo, in barba a tutti gli sforzi di distensione che si stanno facendo in questo scorcio del secondo millennio.

Lo schermo principale di *688 Attack Sub* è tristemente simile a quello del vecchio *Silent Service* della MicroProse: l'interno di una sala comando nella quale, selezionando le varie apparecchiature si accede a una serie di schermate secondarie. Compaiono così sullo schermo la sala radio, quella del sonar, il controllo motori, la santabarbara... ma tutte posseggono una struttura simile e non particolarmente interessante. Probabilmente i pannelli di controllo riproducono le apparecchiature in uso su simili mezzi, tuttavia una successione di pochi interruttori a pulsante disposti sempre secondo lo stesso schema non è certo una visione esaltante.

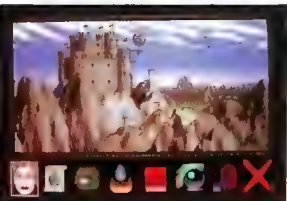
Come in altre simulazioni di questo genere, gran parte dell'azione è rivolta a manovrare il sottomarino in modo da prendere contatto radar con il nemico senza farsi individuare. La cosa non è semplice, anche perché i tempi di manovra del mezzo sono piuttosto lunghi, soprattutto rispetto a quelli dei siluri e delle bombe di profondità. Dimenticate subito l'immagine del capitano che scruta l'orizzonte con il periscopio: le distanze di combattimento sono di gran lunga superiori alla portata di un'apparecchiatura ottica e la navigazione a pelo d'acqua è la manovra più sconsigliabile in assoluto. Lo schema d'azione corretto è purtroppo di una noia mortale. Gli inseguimenti si svolgono spesso oltre la portata del sonar, e si riducono all'analisi dei dati numerici forniti da un paio d'indicatori.

Viene spontaneo dedurre che questo schema di gioco è l'inevitabile conseguenza del tipo di mezzo simulato, ma la precedente esperienza con giochi come *Red Storm Rising* (vedere *Commodore Gazette* numero 6/88) o lo stesso *The Hunt for Red October* suggerisce che un gioco di sottomarini può essere molto emozionante. Se il realismo deve scontrarsi con il divertimento, viva le simulazioni "fantasiose"!

F.R.

DRAGON'S BREATH

Uno schema di gioco singolare in questo parente povero di una simulazione tattica

Computer: Amiga
Supporto: Disco
Prezzo: L. 49.000
Produzione: Palace Software
Distribuzione: Lago (Via Napoleona 16, 22100 Como – 031/300174)

Il gioco è ambientato in una terra medievaleggiante, felice e fiorentissima. L'unica regione ancora devastata da guerre interminabili è proprio quella dov'è ambientato il programma: la zona circostante le pendici del Monte Dwarf. Le leggende locali definiscono il monte "la sede del Male", ma lasciano anche intendere che il segreto dell'immortalità è celato proprio negli anfratti della montagna.

A disputarsi il possesso di un simile tesoro sono tre potenti creature: l'alchimista Bachim, la Bestia Verde Ametrin e la vampira Oureod. Il giocatore controlla uno dei personaggi, mentre gli altri due possono essere affidati alle cure del computer o di altrettanti amici.

Dragon's Breath è diviso in turni della durata di un mese, nel corso dei quali ogni giocatore esegue una serie di scelte mediante un sistema di menu e sotto-menu a icone. Quando tutti i partecipanti hanno compiuto le loro selezioni, l'azione passa al computer che ne mostra gli effetti con sequenze animate di buona qualità. L'obiettivo è recuperare i tre Globi dei Draghi che danno la facoltà di accedere alla camera segreta del Monte Dwarf.

Esaminando la struttura del programma, dotato di "punti di vittoria" e turni di studio tattico, viene spontaneo paragonarlo a una simulazione bellica, della quale in effetti *Dragon's Breath* può essere considerato una sorta di parente povero.

Dal momento che i globi possono essere trovati e manipolati soltanto da un drago, buona parte del gioco consiste nell'allevare un'armata di draghi da impiegare poi nell'azione. A questo scopo esistono opzioni di "ricerca uova", di "incubazione" e di "lancio d'incantesimi". Le modalità d'impiego degli incantesimi sono forse la parte più bella di *Dragon's Breath*. La creazione di pozioni magiche si svolge in una schermata che raffigura un laboratorio da alchimista, e viene effettuata mescolando i numerosi ingredienti acquistati dai mercanti o trovati dai draghi. La cosa interessante è che il "libro degli incantesimi" dà molte indicazioni di massima, ma si guarda bene dal fornire ricette dettagliate per le pozioni. Il giocatore è quindi costretto a fare numerosi esperimenti per identificare gli ingredienti necessari per ogni incantesimo e le loro quantità.

Una volta che si è riunito un esercito di rettili abbastanza numeroso, il gioco entra nel vivo. Ci si può finalmente lanciare in una serie di raid sui territori nemici nel tentativo d'impadronirsi dei globi eventualmente già trovati dagli avversari, o quanto meno per decimarne le schiere. Dal momento che bisogna rimanere in possesso dei tre globi come minimo per un turno, è obbligatorio rimanere con almeno tre draghi guardiani.

La mancanza di azione controllata direttamente dal giocatore rende *Dragon's Breath* consigliabile soprattutto agli amanti dei giochi strategici. Il problema è che, allo stesso prezzo, sono disponibili simulazioni tattiche molto più complesse e divertenti (anche con ambientazioni simili). Gli unici aspetti a favore del gioco sono la buona grafica e il suggestivo sonoro, peraltro rintracciabili anche in altri prodotti più attenti all'aspetto dinamico di queste saghe fantastiche.

Forse mischiare magie, draghi e *quest* oggi non è più sufficiente perché il risultato sia un prodotto interessante. **F.R.**

WARHEAD

Come perdersi nello spazio tra migliaia di lucine lampeggianti

Computer: Amiga
Supporto: Disco
Prezzo: L. 49.000
Produzione: Motion Picture House
Distribuzione: Leader (Via Mazzini 15, 21020 Casciago – 0332/212255)

Certe volte, ripensando al passato e a giochi come *Asteroids* e *Star Raiders*, c'è da chiedersi cosa sia andato storto. Se programmi come quelli erano praticamente perfetti nella loro semplicità – e divertenti – quale miserioso morbo avrà mai potuto provocare l'attuale involuzione dei giochi di combattimento spaziale?

Warhead è un perfetto esempio di questo tipo di fenomeno. Il manuale di gioco occupa undici pagine fittissime di sequenze di tasti, descrizioni di sistemi ed elementi di astrogazione assolutamente indispensabili per riuscire a controllare il gioco, ma una volta che si carica il programma si scopre di essere davanti all'ennesima versione del già citato *Star Raiders*.

Il giocatore interpreta la parte di un pilota di astronavi appena assunto dalle forze militari terrestri. Il nostro pianeta si trova in guerra con una misteriosa e insondabile razza di insettoidi alieni, e l'ultima speranza della razza umana è riposta nell'abilità di alcuni piloti scelti.

Il gioco comincia con una missione di allenamento presso una base stellare in orbita attorno al Sole: si deve solo staccarsi dalla base e ritornarvi. Il fatto che una simile operazione richieda una specifica missione di allenamento la dice lunga sulla manovrabilità del mezzo. Dopo alcuni tentativi alla cieca, destinati a fallire miseramente, l'utente si rivolge al manuale e riesce finalmente a sganciarsi dalla base in una cacofonia di luci e suoni. E qui cominciano i veri problemi.

Come i nostri lettori ben sanno, nello spazio l'attrito ha valori trascurabili, tanto che basta una minima spinta per dare il via a un movimento in grado di protrarsi per milioni di chilometri. Ebbene, *Warhead* si basa proprio su questa legge della fisica. Quando si arriva a mettere in pratica l'istruzione che dice di "spingere in avanti il mouse *leggermente*", si scopre che l'ultima parola non va presa... alla leggera. È sufficiente un movimento di una frazione di millimetro per scagliare la nave in una serie di piroette tanto veloci da trasformare le stelle disegnate sullo schermo in tante strisce luminose. A questo punto gli strumenti diventano inutili, ogni tentativo di riportare le cose alla normalità non fa che aggiungere nuovi vettori al movimento in atto e l'unica possibilità risiede in una sequenza di comandi da tastiera appositamente studiati per stabilizzare il mezzo.

È evidente che con una simile manovrabilità le missioni di combattimento sono un vero dramma, specialmente quando si scopre che i nostri misteriosi avversari riescono a controllare i loro mezzi con una grazia olimpica. In una tale situazione tutti i bellissimi schermi in sovrapposizione, le carte stellari tridimensionali e l'aspetto realmente futuribile della strumentazione perdono d'importanza.

Alla difficoltà di controllo va inoltre aggiunto un notevole grado d'imprecisione nella simulazione delle leggi fisiche: in un prodotto tanto attento ai movimenti inerziali è veramente triste vedere che l'attrazione gravitazionale è praticamente ignorata, così come il "rinculo" provocato dal lancio dei missili. Così *Warhead* finisce per essere non solo un brutto gioco, ma anche una pessima simulazione.

Se tutto ciò che v'interessa è una grafica notevole e una strumentazione da far invidia a quella del Millennium Falcon, *Warhead* può fare al caso vostro. Ma se volete giocare, è meglio rispolverare il vecchio *Star Raiders*. **F.R.**

WORLD CUP KICK OFF

PLAYER MANAGER

Ecco i programmi che ci fanno diventare protagonisti dei mondiali 1990

Computer: Amiga
Supporto: Disco
Prezzo: L. 18.000/25.000 (World Cup) L. 18.000/21.000/39.000 (Player Manager)
Produzione: Reflex (World Cup) Anco (Player Manager)
Distribuzione: Leader (Via Mazzini 15, 21020 Casciago – 0332/212255)

Kick Off non è certo un nome nuovo per gli appassionati di calcio che possiedono un Amiga. Il programma (realizzato da Dino Dini per la Anco) aveva già fatto la sua comparsa sul mercato qualche tempo fa, non riscuotendo in verità un grande successo nel nostro Paese.

I due programmi qui presentati costituiscono comunque delle versioni migliorate e ampliate rispetto al loro predecessore e, per quanto la tecnica di gioco in sé non abbia subito variazioni di rilievo, i due nuovi rampolli della serie hanno tuttavia una propria spiccata e inconfondibile personalità.

World Cup – Kick Off, Franco Baresi. Quale tifoso della nazionale non ha mai sognato di trovarsi in campo a fianco del mitico "piscinin" Franco Baresi? Forse è un sogno impossibile, ma con *Kick Off* può quanto meno essere simulato su un computer. A parte pochi particolari del genere, però, non c'è molto di nuovo in questa versione di *Kick Off*. Il gioco è identico al predecessore tranne che per la possibilità di giocare una simulazione della Coppa del Mondo mettendo in campo una completa serie di squadre nazionali. Si possono creare i gironi a propria scelta, utilizzando le varie nazioni che partecipano a Italia '90, oppure si possono lasciare così come ci vengono presentati dal computer, che ripresenta l'esatta disposizione del mondiale in corso. Il giocatore può poi decidere di controllare le squadre che preferisce durante le eliminatorie che portano alla finale.

Con Player Manager il centro dell'interesse è costituito dagli schemi di gioco

Il gioco d'azione, come già detto, è il vecchio *Kick Off* senza modifiche. Per chi non lo conoscesse, si tratta di un gioco molto rapido, con visione del campo dall'alto, che inquadra uno specchio relativamente piccolo del campo di gioco; nonostante la grafica non eccezionale e i pochi movimenti concessi ai giocatori, il gioco, grazie alla rapidità e semplicità dei comandi e alla notevole velocità, rimane tuttavia impegnativo e divertente. Forse anche più di alcuni altri prodotti molto sofisticati, che alla complessità e all'accuratezza grafica sacrificano spesso quello che è il risultato più importante: la giocabilità globale del prodotto e la manovrabilità degli atleti simulati.

A sinistra: sul tabellone dei mondiali non mancano le imprecisioni. A destra: Kick Off

Un voto positivo è meritatissimo per questo videogioco che, vista anche l'occasione dei mondiali, va preso senz'altro in considerazione se s'intende acquistare un prodotto che ci permetta di continuare a vivere le emozioni di Italia '90 anche davanti ai monitor dei nostri computer. Un piccolo appunto va invece a una traduzione tanto sciatta da non riportare correttamente nemmeno i nomi delle squadre (cfr. Brazile, Rumania e Columbia).

Player Manager. Il gioco, come si può intuire dal nome, ha come nucleo d'interesse la tattica, fuori e dentro il campo. Il programma è molto ben realizzato, e, oltre a offrire tutte le varianti messe a disposizione dai più sofisticati "manager" calcistici in commercio, permette all'aspirante allenatore di vedere o giocare, sempre tramite *Kick Off*, le partite da lui impostate.

L'esordiente allenatore è un gioca-

tore di classe internazionale al quale è stata affidata una squadra di serie C. Deve utilizzare tutta la sua abilità in campo e negli spogliatoi per riportare la sua squadra nella massima divisione. Per raggiungere l'obiettivo deve fare acquisti oculati (vi sono più di mille giocatori nel campionato, e ciascuno ha proprie caratteristiche) aiutato da schede dettagliatissime fornite dal computer. Deve sfruttare le tattiche di gioco registrate sul disco oppure proporne di sue, creandole tramite un'apposita funzione grafica. Deve dare le direttive per gli allenamenti, decidendo magari di far allenare alcuni giocatori in ruoli diversi da quello naturale. La parte d'azione

Player Manager nella sua parte d'azione

del programma è invece rappresentata da una versione leggermente modificata di *Kick Off*, che ne mantiene però invariata la velocità, i controlli e il tipo di gioco.

Se l'allenatore seleziona se stesso, può anche prendere parte al gioco personalmente, e cambiare giocatori e tattiche durante la partita. Si può comunque vedere rapidamente il risultato della propria impostazione tattica osservando su uno scanner la versione accelerata della partita.

Nel corso del gioco c'è una vera e propria messe di dati di cui si deve tener conto, che contribuiscono a dare la sensazione di avere a che fare con una vera squadra, con tutti i suoi problemi tattici, finanziari e anche umani.

Nel complesso, dunque, due prodotti molto ben realizzati soprattutto dal punto di vista della giocabilità. Pur non essendo dotati di caratteristiche particolarmente eclatanti, ci sembra che la dote principale che si richiede a un gioco – cioè la capacità di dare divertimento – non manchi a nessuno dei due. **N.F.R.**

E-MOTION

Con questo programma la fisica diventa un gioco

Computer: C-64/128/Amiga
Supporto: Cassetta/Disco
Prezzo: L. 18.000/25.000
Produzione: U.S. Gold
Distribuzione: Leader (Via Mazzini 15, 21020 Casciago – 0332/212255)

Tetris, pubblicato un paio d'anni fa in versione per Commodore 64 e subito convertito in ogni formato possibile (coin-op compreso), è senza dubbio uno dei programmi che hanno fatto più proseliti nell'industria dei videogiochi degli ultimi anni. Il programma, sviluppato nell'est europeo, è riuscito a dimostrare sul campo – con l'incredibile quantità di copie vendute – quali sono le qualità più apprezzate dagli utenti di computer in un programma ludico: uno schema di gioco semplice, in cui sia magari necessario un minimo d'impegno intellettuale.

Sulla scia di *Tetris* sono apparse decine di giochi definibili come "intellettuali-astratti". Negli ultimi tempi abbiamo ammirato perle come *Block-Out* e quasi-plagi come *Coloris* accomunati nella famiglia degli "eredi di *Tetris*", ma mentre la maggior parte delle software house si affannava a pubblicare giochi a base di cubetti colorati, alcuni gruppi di programmatori stavano sfruttando con più intelligenza la loro creatività.

Il protagonista dell'azione è un piccolo veicolo di forma triangolare ingabbiato all'interno di una sfera colorata. All'inizio di ogni livello, sullo schermo compaiono altre sfere statiche che pulsano di un'indefinita energia interna.

Per vincere, il giocatore deve far collidere le palline di uguale colore in modo da farle collassare e quindi sparire dal piano di gioco. Per spostare le particelle deve urtarle con la sua sfera direzionabile, che ha però, come tutte le altre componenti del gioco, la tendenza a subire le forze inerziali. La sensazione generale è di essere tornati nel passato, davanti a un coin-op del gioco *Asteroids*, anche a causa della forma della navicella.

La prima difficoltà è rappresentata dal fatto che, facendo urtare sfere di diverso colore, si provoca la creazione di una nuova pallina che andrà a sua volta eliminata. Non appena compaiono, queste sfere secondarie possono essere assorbite semplicemente passandovi sopra con la navicella: in questo modo si ottiene anche una certa quantità di energia supplementare, indicata sullo schermo sotto forma di una barra colorata. L'energia extra svolge una funzione vitale nel caso che le sfere esplodano, assorbendo buona parte dell'energia del giocatore. Per prevenire le esplosioni si deve cercare di far sparire rapidamente le particelle, ma è più facile a dirsi che a farsi.

La versione Amiga (sopra) e C-64 (sotto)

Un altro problema da affrontare in *E-Motion* è rappresentato dalle barriere fisse e mobili presenti in alcuni livelli, che complicano la situazione ostacolando le manovre della sfera. La terza e ultima difficoltà prende forma nei legami elastici che spesso uniscono i vari elementi, compreso il veicolo controllato dal giocatore. Per capire esattamente di che cosa stiamo parlando dovreste proprio dare un'occhiata al programma.

Una rapida descrizione non può rendere giustizia alla varietà di situazioni che si presenta durante una partita di *E-Motion*. Tutto quel che possiamo fare è segnalare la possibilità di giocare in due e... suggerirvi di fare un salto nel vostro negozio di fiducia e chiedere di provarlo. **F.R.**

MOONSHADOW

Il primo, divertente gioco di una nuova software house italiana

Computer: C-64/128
Supporto: Cassetta/Disco
Prezzo: L. 18.000/25.000
Produzione: Idea
Distribuzione: S.C. (Via Mazzini 12, 21020 Casciago - 0332/222052)

Potrà essere un'osservazione banale, ma è bello vedere che gli italiani si sono finalmente dati da fare anche nel campo del software ludico. La Idea è una delle più giovani software house del nostro Paese, ed è stata fondata con il preciso scopo di creare videogiochi concorrenziali con la produzione inglese (il 90 per cento dei giochi per home computer viene attualmente prodotto nel Regno Unito). Per ora ha all'attivo due titoli, entrambi per computer Commodore: il primo è *Bomber Bob*, uno shoot'em up per Amiga, e il secondo è questo *Moonshadow*, programma per C-64 realizzato interamente dal giovane Paolo Galimberti.

In un recente incontro, Paolo ci ha rivelato il suo amore per la letteratura fantasy e per i giochi a essa ispirati, primo fra tutti quel *Sacred Armour of Antiriad* della Palace Software che ebbe tanto successo qualche anno fa.

In *Moonshadow* queste passioni del programmatore vengono rivelate da parecchi particolari, a partire dalla trama che sembra essere un misto del tolkieniano *Silmarillon* e della saga di Dragonlance. Il protagonista è infatti un guerriero senza nome che, scelto dagli dei del Bene, deve impedire il risveglio di un diabolico Grande Serpente. La creatura delle tenebre uscirà dalla sua tana sotterranea quando la luna nera Septerios avrà eclissato la luna luminosa Fulax, facendo precipitare il pianeta nell'oscurità. Purtroppo non abbiamo spazio sufficiente per approfondire l'affascinante background del gioco, che combina numerosi elementi delle più disparate mitologie con qualche frammento di magia nera.

Con grande gioia degli ancora numerosi possessori di unità a nastro, il programma consta di un unico blocco, cui è aggiunta un'apprezzabile schermata di caricamento. Questo formato limita i caricamenti a una sola operazione, creando una certa nostalgia per i tempi in cui il problema delle fastidiose "pause per caricamento" era ancora sconosciuto.

Il gioco si presenta come un'avventura dinamica con visione laterale. Il mondo di *Moonshadow* appare come una serie di schermate a sostituzione, unite sia verticalmente, sia orizzontalmente. Già all'apparire della prima immagine la mente torna al citato *Sacred Armour of Antiriad*: l'autore del gioco si è dichiaratamente ispirato a quel classico, e benché gli elementi che compongono le schermate in effetti non si possano definire copiati, la somiglianza potrebbe infastidire qualche giocatore.

Una nota di merito è costituita dall'aspetto sonoro del gioco, dotato di alcune musiche davvero notevoli. La loro qualità non ha niente da invidiare ai lavori degli anglosassoni Whittaker o Maniacs of Noise, e nel complesso ricorda le colonne sonore composte dal primo Martin Galway, forse il miglior autore di musiche per home computer in assoluto. La cosa stupefacente è che anche questo aspetto sia stato curato personalmente e senza aiuti esterni da Galimberti, che per hobby suona la chitarra in un gruppo heavy metal.

Veniamo ora alla struttura del gioco. In *Moonshadow* ritroviamo uno degli schemi più collaudati del mondo dei videogiochi. Il protagonista deve infatti aprirsi la strada lungo le schermate correndo, saltando e abbassandosi: premendo il pulsante di fuoco viene lanciata l'arma in uso (fra le quattro o cinque disponibili), con la quale è possibile eliminare la maggior parte degli avversari che popolano la schermata. I micidiali nemici hanno le forme più varie, e vanno da comuni pipistrelli tentacolati a micidiali dragoni sputafuoco, fino a giganteschi bulbi oculari rimbalzanti. Il contatto con i nemici o con i loro colpi riduce l'energia vitale dell'eroe, che perde la sua unica vita quando l'energia arriva a zero.

Per recuperare energia, e risolvere la maggior parte dei problemi presentati dal gioco, si devono selezionare e impiegare sei oggetti trasportabili contemporaneamente, fra i quali abbondano le "pozioni rigeneratrici". La difficoltà viene aumentata dal breve lasso di tempo che precede l'eclissi, presentata sotto forma grafica in un'apposita finestra.

La sua grande giocabilità rende *Moonshadow* una delle avventure dinamiche più divertenti fra quelle uscite negli ultimi tempi in Europa. Un titolo che ci sentiamo di raccomandare e che rappresenta davvero un ottimo inizio per la Idea e per Paolo... ma già si parla di un nuovo gioco che sarà sicuramente una grossa sorpresa!

F.R.

WAR OF THE LANCE

Una simulazione tattica basata sulla saga di Dragonlance

Computer: C-64/128
Supporto: Disco
Prezzo: L. 59.000
Produzione: SSI
Distribuzione: Leader (Via Mazzini 15, 21020 Casciago – 0332/212255)

Questa recensione è la prima di una serie dedicata ai giochi per computer basati sulla saga fantasy di *Dragonlance*. Il ciclo è composto di una dozzina di romanzi scritti da numerosi autori, tra i quali spiccano ovviamente i creatori dell'idea, Margaret Weis e Tracy Hickman. La grande diffusione dei romanzi (editi in Italia da Armenia) ha causato una sorta di reazione da cui sono nati innumerevoli "spin-off" della serie, non ultimi diversi programmi ludici.

Non è facile riassumere la trama del ciclo per sommi capi: basti sapere che la vicenda è ambientata sul pianeta Krynn, attorno al quale orbitano tre lune che rappresentano gli dei dell'Ordine, del Caos e della Neutralità. Krynn si trova in una fase in cui l'influsso del dio Paladine (Ordine), abbandonato dai suoi fedeli, è quasi scomparso. La dea malvagia Takhisis (Caos) controlla truppe sparse su tutto il pianeta, ed è favorita nei suoi piani di conquista dalla presenza nelle sue fila di creature seminvulnerabili che tutti credevano estinte: i draghi. Una minaccia terribile per le pacifiche civiltà degli umani, degli elfi, degli gnomi e dei piccoli kender.

Le sorti di quella che ormai è diventata una guerra dichiarata fra Bene e Male volgono però a favore delle razze più pacifiche quando vengono scoperte le Dragonlance, armi magiche capaci di uccidere i draghi, e soprattutto quando entrano in campo i Draghi Metallici, una razza di leviatani votati all'Ordine che aiuteranno i numerosi eroi della vicenda.

War of the Lance può essere definito come il "bigino" di *Dragonlance*: laddove gli altri giochi si preoccupano di approfondire l'uno o l'altro aspetto della saga seguendo le sorti di questo o quell'altro gruppo d'eroi, questo programma tratta la storia nella sua totalità, non scendendo mai troppo nei dettagli. Il gioco è presentato come una normale simulazione tattica (o war-game, per gli anglofoni) ambientata nel continente di Ansalon, dove il Bene è impersonato dalle forze di Whitestone e il Male da quelle dell'Highlord. Possono prendere parte al gioco una o due persone, ed è possibile scegliere se affrontare tutta la Guerra delle Lance dagli inizi o se concentrarsi soltanto su una delle sue molte fasi. In ogni caso, si possono sempre definire degli handicap, moltiplicando il numero dei livelli di difficoltà.

Gli autori del gioco hanno evidentemente studiato con attenzione i romanzi, e sembra tutto perfetto, a partire dalla mappa di gioco per finire con i personaggi e le situazioni. L'azione di gioco segue piuttosto da vicino quella di ogni altro prodotto della SSI: ogni turno è composto di 19 sottosezioni, e una partita completa comporta come minimo una cinquantina di turni, portando così la durata complessiva di una "guerra" a diversi giorni di gioco.

Oltre alle immancabili fasi di movimento e combattimento, questa simulazione tattica è caratterizzata dalla presenza di due fasi particolarissime. La prima riguarda i rapporti diplomatici con le molte nazioni di Ansalon, che devono essere convinte a collaborare con la propria fazione e a fornire truppe e aiuti di vario genere. La seconda fase deve la sua origine alla trama dei libri, e vede il giocatore coordinare gli sforzi di una serie di "campioni" impegnati nella ricerca di oggetti particolari: armi e armature, oppure i manufatti mistici necessari per la creazione delle Dragonlance. Simili sottosezioni sono il vero punto di forza del gioco, che altrimenti non sarebbe distinguibile da un comune war-game.

Sotto l'aspetto grafico e sonoro il programma non offre nulla di memorabile, se non la possibilità di esaminare la mappa in tre diversi livelli di dettaglio, di cui uno disponibile soltanto durante i combattimenti. L'unico vero difetto è in realtà inestricabilmente legato alla sua stessa natura di simulazione tattica. Parliamo dei lunghissimi tempi d'azione: un turno completo richiede in media mezz'ora, con una crescita geometrica dei tempi a mano a mano che alle nostre forze si uniscono sempre più truppe.

War of the Lance è sicuramente consigliabile a tutti i fan dei romanzi, che vi troveranno i loro eroi e apprezzeranno al meglio avvenimenti come gli attacchi paralizzanti dei draghi o l'inefficienza delle macchine da guerra degli gnomi. Gli appassionati di simulazioni tattiche saranno affascinati dalle fasi diplomatiche e di ricerca, mentre i neofiti apprezzeranno in particolare le meccaniche di gioco originali ma relativamente semplici, una caratteristica rarissima in prodotti analoghi. È un buon modo per fare conoscenza con Krynn, anche se una lettura dei romanzi aiuterebbe ad apprezzare la trama e l'azione dell'avventura.

F.R.

SCATTA IL CRONOMETRO PER L'AMIGA 2000

Tre nuove possibilità di accelerare il vostro computer: le schede A2630 della Commodore, Hurricane 2800 della Imtronics e Impact A3000/4000 della GVP

di Sheldon Leemon

Semplicemente aggiungendo al sistema una scheda 68020, gli amanti delle alte prestazioni sono già riusciti a far funzionare i programmi a una velocità due o tre volte superiore a quella dell'Amiga di serie. Ora è nata una famiglia di schede 68030 che permette di raggiungere velocità due o tre volte superiori a quelle consentite dalle 020... il che significa moltiplicare per dieci la velocità di lavoro dell'Amiga 2000!

Queste straordinarie schede, che mettono l'Amiga alla pari con i più veloci personal computer, sono la A2630 della Commodore, la Hurricane 2800 della Imtronics e le due Impact A3000 e A4000 della Great Valley Products.

Ciascuna di queste schede si inserisce nello slot coprocessore a 86 pin dell'A2000, e aumenta le prestazioni del computer in parecchi modi. Per dirne una, il 68030 può funzionare a frequenze di clock molto superiori a quelle del 68000 o del 68020. La Hurricane 2800 funziona a 28 MHz, una frequenza di lavoro doppia rispetto a quella delle schede basate sul 68020, e quadrupla rispetto a quella del microprocessore standard dell'Amiga 2000; la Impact funziona invece a frequenze di clock variabili fra 25 e 33 MHz, e la A2630 opera normalmente a 25 MHz (ma questo valore può anche aumentare).

Come il 68020, il 68030 ha un cache-buffer/istruzioni di 256 byte, grazie al quale è possibile eseguire brevi loop nella memoria privata del microprocessore, evitando di perdere tempo a localizzare e caricare le istruzioni. In aggiunta, possiede un cache-buffer/dati di 256 byte che permette di accedere ai dati di uso più frequente senza doverli estrarre dalla memoria principale.

L'impiego di cache-buffer/dati accelera in modo significativo l'esecuzione dei programmi, ma a volte causa problemi con i dispositivi DMA (Direct Memory Access, memoria ad accesso diretto), compresi i controller per hard disk DMA. Il problema di base è che la CPU non può sapere quando c'è un cambiamento nel contenuto della memoria di un dispositivo DMA. Di conseguenza, se i dati immagazzinati nel cache-buffer (o memoria-cache) si trovano alla locazione 100.000, e un dispositivo DMA cambia il contenuto di questa locazione, la CPU continua a considerare validi i vecchi dati contenuti nella sua memoria personale e non legge quelli nuovi dalla memoria principale. Benché una simile evenienza non sia molto probabile, conduce comunque a una possibilità di errore. Per evitarla, tutte le schede contengono "protezioni" hardware che impediscono d'immagazzinare nella memoria-cache dati contenuti nella chip RAM (ovvero dati che possono essere alterati da dispositivi DMA come il Blitter). Ma chi possiede un hard disk DMA sappia che potrebbe essere preferibile non utilizzare i cache-buffer/dati.

Il 68030 ha una propria unità di gestione della memoria (o MMU, da Memory Management Unit), utilizzabile dall'utente per rilocare le ROM del Kickstart nella memoria a 32 bit, riducendo il tempo d'accesso alle routine contenute in ROM. Il coprocessore matematico in virgola mobile 68882, che in queste schede è sempre opzionale, può svolgere complesse operazioni matematiche a una velocità che supera di decine di volte quella raggiungibile da qualsiasi routine software. Si osservi, però, che la maggior parte dei programmi (come per esempio i word processor, i programmi di comunicazione e i giochi) non ricorrono a calcoli matematici in virgola mobile e quindi non possono approfittare della presenza del 68882. Un chip matematico, invece, accelera da dieci a cento volte applicazioni come i CAD, i fogli elettronici e i programmi di grafica 3D.

Le differenze tecniche

Malgrado abbiano alcuni aspetti in comune, queste schede presentano anche alcune significative differenze. Seguendo il modello della 68020, la Hurricane 2800 impiega un disegno sincrono, il che significa che può funzionare soltanto a un multiplo pari della frequenza di clock del 68000 (7,1 MHz); nel caso specifico, la frequenza è di 28,4 MHz. La A2630 e le due schede Impact della

GVP, invece, sono asincrone, e teoricamente possono funzionare a qualsiasi frequenza compresa in un certo intervallo (la Commodore però ha messo in commercio soltanto una versione a 25 MHz della A2630).

Le prime versioni delle schede GVP impiegavano chip 68030 a 16 MHz, ma dal momento che i microprocessori ad alta velocità costano sempre meno e sono sempre più facilmente disponibili, ora la GVP ha messo in commercio versioni delle sue schede che funzionano a 25, 28 e persino a 33 MHz. Variare la velocità operativa della scheda significa semplicemente installare un processore più rapido e l'appropriato cristallo oscillatore.

In realtà è possibile aumentare la velocità operativa delle schede Impact e A2630 semplicemente cambiando il cristallo di clock (quello contenuto nella Impact è montato su zoccolo, mentre quello della A2630 è saldato sulla scheda). Facciamo comunque notare che far funzionare un 68030 a una velocità superiore a quella prevista è una scelta quantomeno controversa. Chi è favorevole a questa soluzione, chiamata "over-oscillation" (super-oscillazione) o "over-clocking", osserva che le velocità operative dei chip non sono così nette e offrono ampi margini di sicurezza. I più prudenti ribattono che un margine è necessario, perché far funzionare un chip troppo in fretta può causare surriscaldamenti e malfunzionamenti. La possibilità di prestazioni non affidabili è sufficiente per scoraggiarli dall'idea di "mettere sotto pressione" i chip. Inoltre, anche se una maggior velocità di clock fa decollare le prestazioni, l'aumento di velocità non è lineare: far funzionare una scheda a 33 MHz non significa necessariamente ottenere una velocità doppia rispetto a una scheda a che funziona 16 MHz. Un fattore limitativo è la velocità con cui il processore accede alla memoria; in altre parole, a meno che non possediate chip di memoria altrettanto veloci, non riuscirete a sfruttare fino in fondo i vantaggi di una frequenza di clock superiore.

Che siate pro o contro l'over-clocking, sappiate che la Hurricane 2800 è equipaggiata con un 68030 da 25 MHz che funziona a 28 MHz, mentre la GVP impiega un chip da 25 MHz che funziona a 25 MHz per il modello A3000 e un chip da 33 MHz che funziona a 33 MHz per il modello A4000 (ma è imminente l'adozione di schede da 28 MHz al posto di quelle da 25 MHz). La A2630 è disponibile soltanto con un processore da 25 MHz che funziona a 25 MHz. Il coprocessore matematico della Hurricane ha una frequenza di clock fissa di 28 MHz, mentre le schede della Commodore e della GVP possono entrambe essere equipaggiate con coprocessori matematici funzionanti a un clock diverso da quello della CPU, utilizzando cristalli diversi.

Al posto di guida

Sia la Impact che la Hurricane contengono un'interfaccia per l'hard disk, ma la soluzione adottata è diversa. La scheda Hurricane 2800 fornisce la familiare interfaccia SCSI, comunemente utilizzata per collegare fino a sette hard disk all'Amiga. Non siamo comunque in grado di valutarne le prestazioni, perché il driver software non era ancora definitivo nel momento in cui è stata scritta questa recensione. L'interfaccia per hard disk della GVP, invece, l'abbiamo osservata in funzione per un certo tempo: include un bus stile-AT che permette il collegamento con due hard disk IDE (Integrated Drive Electronics), progettata per l'uso con gli AT IBM. La Great Valley Products offre combinazioni di hard disk più scheda acceleratrice, che includono un hard disk Quantum AT (lo stesso scelto dalla Commodore per i suoi PC-40 III). Le ROM per il boot che si collegano alla scheda Impact consentono l'autoboot dall'hard disk della scheda, facendo uso del Fast File System. L'accluso software d'installazione rende molto semplice creare le partizioni dell'hard disk e installarvi il *Workbench*.

L'uso di RAM a 32 bit rende la scheda ancor più efficiente perché elimina il collo di bottiglia che si crea quando il 68030 deve dialogare con una memoria a 16 bit come la chip RAM, oppure con i chip dedicati. E infatti tutte e tre le schede sono in grado di sfruttare una memoria a 32 bit (ma soltanto la A2630 mette a disposizione uno spazio per questa RAM sulla scheda principale). Per aggiungere memoria di quel tipo alla Hurricane 2800 oppure alla Impact, si deve collegare alla scheda acceleratrice una scheda di memoria aggiuntiva. Il risultato è una specie di panciuto sandwich formato dalla sovrapposizione di due schede, ma fornisce all'utente una flessibilità di configurazione molto maggiore: si può cominciare acquistando la sola scheda acceleratrice e in seguito, quando le proprie disponibilità economiche lo consentono, si può sempre aggiungere la scheda di memoria. La Hurricane 2800 utilizza la stessa scheda di memoria impiegata dalle precedenti schede della serie Hurricane, e nello zoccolo trovano posto sia la chip RAM da 1 megabit (256 x 4) sia quella da 4 megabit (1024 x 4). Si può aggiungere memoria 1 megabit alla volta utilizzando i chip più piccoli, oppure 4 megabit alla volta utilizzando quelli più grandi, fino a raggiungere un massimo rispettivamente di 4 o di 16 megabit. La scheda di memoria della GVP adotta un indirizzamento nibble-mode, che permette un più rapido accesso alla memoria ma richiede che i singoli incrementi siano di quattro megabit. La scheda GVP impiega SIMM nibble-mode da 8 megabit x 1 bit (da non confondere con i SIMM page-mode utilizzati nei computer Macintosh e PS/2) per aggiungere quattro oppure otto megabyte di RAM a 32 bit.

Con la A2630 della Commodore le scelte sono un po' più limitate: contiene 2 megabyte di RAM a 32 bit, e anche se questa dotazione può essere aumentata fino a 4 megabyte non si tratta della semplice operazione d'inserire un chip in uno zoccolo, come con le altre schede. È necessario saldare alla scheda ben 16 chip RAM (256 x 4 zip)... un lavoro che è meglio lasciare al "personale tecnico qualificato". Un altro inconveniente della scheda Commodore è che i suoi chip di memoria sono relativamente lenti (100 nanosecondi). La scheda sarebbe più rapida se i suoi componenti fossero da 80 nanosecondi, ma non si può approfittare di questa caratteristica a meno che non si dissaldino e non si sostituiscano i chip di memoria originali. Come le altre schede, la A2630 possiede un connettore che la collega a un bus a 32 bit, lasciando aperta la possibilità di più veloci espansioni di memoria che superino il limite dei quattro mega.

Benvenuti nel sistema

Le schede di memoria della A2630 e della Impact si autoconfigurano entrambe (in altre parole, il computer riconosce automaticamente la loro presenza al momento del boot, e colloca la memoria supplementare da

esse fornita al primo posto nella lista degli otto megabyte riservati alle espansioni di memoria). La memoria fornita dalle due schede può essere sfruttata da dispositivi DMA, cosa che dovrebbe accelerare l'accesso da parte dei controller di hard disk e dispositivi simili.

La scheda di memoria della Hurricane, invece, non si autoconfigura, e quindi si deve mandare in esecuzione un apposito programma di configurazione per aggiungere la memoria al sistema e per copiare il codice del Kickstart nella memoria a 32 bit. In genere anche qui la memoria viene aggiunta negli otto mega destinati alle espansioni, ma può anche essere configurata all'esterno di questo spazio, permettendo quindi di raggiungere un massimo di nove megabyte di espansione. Mandare in esecuzione il programma di configurazione, comunque, non è un grosso sforzo: è infatti sufficiente aggiungere il comando alla startup-sequence. Inoltre non è necessario rimandarlo in esecuzione durante un eventuale warm-boot. Comunque la mancanza di autoconfigurazione da parte della scheda non ha altre conseguenze. A differenza delle schede acceleratrici della GVP e della Commodore, la Hurricane 2800 non funziona con il programma *SetCPU* di Dave Haynie; le altre due schede invece se ne servono per controllare i cache-buffer, rilocare il Kickstart e consentire l'uso del disco Kickstart per il boot di versioni alternative del sistema operativo. Il programma di configurazione copre le due prime funzioni ma non permette di avviare la macchina da un disco Kickstart.

Da un punto di vista pratico, i chip 68030 contenuti nelle schede acceleratrici sostituiscono in tutto e per tutto il 68000, ma è tuttavia possibile disabilitare temporaneamente il 68030 e utilizzare il 68000 per mandare in esecuzione quei rari programmi (giochi protetti contro la copiatura, per la maggior parte) che con il 68030 non funzionano. Nel caso della Impact e della Hurricane l'operazione si svolge azionando uno switch hardware che l'utente deve comperare e innestare su una coppia di pin nella scheda. Con l'interruttore in una posizione il boot viene effettuato impiegando il 68000, mentre nell'altra posizione viene impiegato il 68030. La scheda A2630 della Commodore ricorre a una tecnica più elaborata: se si tengono premuti entrambi i pulsanti del mouse al

LA PROVA IN CIFRE

Sono state eseguite tre categorie di test, impiegando dapprima applicazioni che non facevano uso di matematica in virgola mobile. I test di velocità della RAM (Fast RAM/Chip RAM/RAM Speed) sul disco CSA servono per valutare i tempi di esecuzione delle istruzioni eseguite da fast RAM e da chip RAM, senza la presenza di cache-buffer/istruzioni. Il programma *Nsieve* è della CSA, mentre gli altri programmi "sieve" (ovvero di valutazione) come *Sieve/100* e *CPU/Memtest* sono della Ronin, responsabile anche dello sviluppo della scheda Hurricane. Questi test mandano in esecuzione alcune variazioni di un famoso programma che individua i numeri primi (un test per valutare la velocità in operazioni non in virgola mobile). Il test di ricerca/sostituzione illustra il tempo richiesto per realizzare 800 operazioni di ricerca e sostituzione in un file di testo di 96 mila caratteri, utilizzando l'editor *TxEd* della MicroSmiths. Il test *Remap* mette in luce quanto tempo è necessario a *Deluxe Paint II* (Electronic Arts) per trasformare un'immagine di 640 x 400 pixel a 16 colori in un'immagine a due colori. *WritePixel* cronometra il tempo necessario per riempire un rettangolo utilizzando la funzione WritePixel del ROM Kernel. Infine, *Speed* è un benchmark progettato recentemente da Jez San per confrontare con la A2620 le schede acceleratrici fornite di 68030.

La seconda classe di test illustra gli aumenti di velocità in operazioni che fanno uso di routine software in virgola mobile. I due test *Savage* forniscono i risultati di un famoso benchmark in virgola mobile, mandato in esecuzione dai programmi CSA e Ronin. Anche il programma *MandelTest* viene dal disco CSA, e mostra il tempo necessario per tracciare l insieme di Mandelbrot un grafico che richiede calcoli comprendenti oltre due milioni di operazioni in virgola mobile. Il test *WorldMap* mostra il tempo richiesto per tracciare una mappa della Terra sulla base di circa 5 mila coordinate e uno schema di proiezione ortografica (in questo programma il rapporto tempo-di-tracciamento/tempo-di-calcolo è maggiore che nel *MandelTest*).

Il terzo insieme di test mostra l'incremento di velocità di vari programmi contenenti calcoli in virgola mobile, eseguiti con l'ausilio del coprocessore matematico. Sono quasi tutti basati sui test *Fast Floating Point* e utilizzano il 68881 sia direttamente sia tramite le librerie matematiche IEEE dell'Amiga. Il test conclusivo di questa categoria mostra il tempo necessario per eseguire il rendering dell'immagine in 3D di una tazzina da tè in modo HAM 320 x 400, utilizzando la versione in virgola mobile di *Turbo Silver* della Impulse. Nella prova, tutte le schede acceleratrici utilizzavano il chip matematico 68882 funzionante alla stessa velocità del microprocessore principale. Tutte utilizzavano inoltre RAM a 32 bit, anche se il tempo di accesso alle varie chip RAM non era costante (la A2630 era fornita di RAM da 100 ns, la Impact di RAM da 80 ns e la Hurricane di RAM da 70 ns).

Test non in virgola mobile		***A2630***	***GVP25***	***GVP28***	***H2800***	***GVP33***
RAMtest Fast	sec	2,04	2,08	1,90	1,52	1,58
RAMtest Chip	sec	9,52	11,06	10,46	7,91	7,98
RAMspeed	MHz	20,39	20,39	20,67	26,72	25,00
Nsieve (CSA)	sec	0,27	0,27	0,23	0,23	0,20
Nsieve (CSA)	guadagno	8,61	8,76	10,01	10,79	11,81
WritePixel (CSA)	sec	3 98	3,66	3,34	3,34	3,10
WritePixel (CSA)	guadagno	3 60	3,90	4,20	4,30	4,60
Sieve (Ronin)	sec	5,70	5,66	5,06	4,76	4,25
CPU/Memtest (Ronin)	guadagno	10 62	10,62	11,50	12,55	14 53
Ric /Sost	sec	58,40	57,00	48 20	44,90	46 60
Remap	sec	6,00	5,90	5,60	5,60	5,50
Speed	tick	116,00	96,00	85,00	89,00	71 00
Speed	guadagno	2,01	2,43	2,75	2 62	3,29
Test in virgola mobile						
Savage (CSA)	sec	6,30	5,90	5,32	4,94	4,34
Savage (CSA)	guadagno	8,71	9,27	10,28	11,07	12,60
Savage/FFP (Ronin)	sec	1,53	1,38	1,22	1,20	1,03
Mandeltest 68000	sec	53,58	48,48	44,54	42,88	37,74
Worldmap	sec	7,90	6,60	6,10	6,10	5,21
Test in virgola mobile con il 68882						
Savage/020 (CSA)	sec	0,22	0 22	0,19	0,20	0 15
Savage/IEEE (Ronin)	sec	0,43	0 42	0 38	0,38	0 30
Savage/881 (Ronin)	sec	0,22	0,22	0 19	0 18	0 15
Mandeltest	sec	11,92	11 56	10 40	10,32	9 04
Mandeltest	guadagno	35,40	36 50	40 60	40,90	46 70
Worldmap/IEEE	sec	5 30	5 30	4 75	4 60	3 70
Worldmap/882	sec	2 75	2 45	2 23	2 23	2 05
Turbo Silver	min sec	61 04	36 38	33 52	39 46	31 26

momento del boot, appare un menu che permette di scegliere l'AmigaDOS del 68000, l'AmigaDOS del 68030 oppure l'Amix (la versione per Amiga dello Unix). Malgrado il fatto che la GVP non dia supporto a questo meccanismo di selezione, la società dichiara che eseguendo il boot con le opportune ROM, anche la sua scheda permette la selezione via software dello Unix.

Quando si esclude il 68030, la scheda restituisce una piena compatibilità software ma nel contempo viene disabilitato qualunque altro dispositivo a essa collegato, come per esempio la memoria a 32 bit o il coprocessore matematico 68882. Allo stesso modo, l'interfaccia hard disk della GVP non funziona quando è escluso il 68030. La Imtronics, invece, dichiara che la sua interfaccia hard disk SCSI è in grado di operare anche in modo 68000.

Ma quanto sono veloci?

L'aumento di velocità ottenibile con queste schede varia da applicazione ad applicazione, anche se qualche miglioramento si nota sempre. A parte la matematica in virgola mobile, queste schede svolgono molte altre operazioni a una velocità da cinque a nove volte superiore a quella di un Amiga non accelerato. L'incremento di velocità sarà più notevole per le applicazioni che in condizioni normali sono lente, come quelle di desktop publishing e di CAD, e nelle situazioni in cui il multitasking tende a far "impantanare" la macchina. Quando si usa un 68030 in coppia con una memoria a 32 bit si nota subito, in quasi tutte le applicazioni, una netta diminuzione del tempo d'aggiornamento delle finestre.

I test che abbiamo condotto dimostrano, come c'era da aspettarsi, la superiorità delle schede con frequenza di clock più alta e memoria a tempo d'accesso più breve. Con una frequenza di clock di 25 MHz e impiegando una RAM da 80 ns, la scheda Impact della GVP si è dimostrata più veloce della A2630, che impiega una RAM da 100 ns. A 28 MHz e con RAM da 70 ns, la Hurricane 2800 si è dimostrata leggermente più veloce della GVP con RAM da 80 ns. Ma tra le varie configurazioni prese in esame, la più veloce è quella costituita dalla Impact a una frequenza di clock di 33 MHz.

Gli stessi elementi che avevano decretato il trionfo della scheda A2620, tra tutti gli acceleratori basati sul 68020, ci costringono ora a classificare la A2630 all'ultimo posto: mettere in commercio una sola configurazione avrebbe avuto senso soltanto se la scheda fosse stata dotata di una e una sola velocità di processore... ma non con una scheda progettata secondo modalità asincrone! E anche il fatto che la scheda abbia di serie le relativamente lente RAM da 100 ns va a scapito delle sue prestazioni. I test mostrano che la A2630 è un po' più lenta delle altre schede in quasi tutte le categorie, ma se la cava particolarmente male nel test di ray-tracing (abbiamo utilizzato la versione per 68020 di *Turbo Silver* della Impulse) nel quale è stato effettuato il rendering dell'immagine di una tazzina da tè. Mentre le altre schede hanno portato a termine l'elaborazione in un tempo pari a circa la metà di quello necessario alla A2620, la A2630 ha superato il suo predecessore soltanto del 15 per cento.

Se la vostra più grande ambizione è la velocità, la A2630 non è la scelta più felice.

Come la sua controparte per il 68020, la Hurricane 2800 ricompensa in abbondanza l'acquirente. Per un prezzo decisamente inferiore a quello della A2630, offre prestazioni più veloci, una memoria più facile da espandere, e un'interfaccia SCSI per il boot. Vi sono tuttavia alcuni fattori su cui è necessario riflettere bene prima dell'acquisto. Dal momento che l'interfaccia SCSI non è ancora operativa, non ci sono chiare indicazioni sul suo modo di operare. Lo schema sincrono della scheda, inoltre, la rende più sensibile delle altre alle temporizzazioni del sistema e la memoria non-autoconfigurante la qualifica come leggermente fuori-standard. Anche se ci sono particolari posizioni dei jumper che aiutano ad adattare la scheda al proprio sistema, mi è capitato di trovare un A2000 nel quale la scheda non voleva saperne di funzionare. Probabilmente la Hurricane 2800 farà presa soprattutto sugli utenti interessati alle migliori prestazioni ai prezzi più bassi.

Quella della GVP è, in ordine di tempo, la prima scheda acceleratrice 68030 per l'Amiga, ed è già stata abbondantemente messa alla prova. Dal tempo della release iniziale, la società costruttrice ha individuato e corretto tutta una serie di piccoli errori di progettazione, e ora la scheda funziona in modo quasi impeccabile. Inoltre, anche se le prime versioni dell'interfaccia hard disk erano piuttosto lente, ora l'interfaccia s'inserisce di diritto tra le più rapide, grazie a una velocità di lettura che arriva a 650K al secondo. Nella sua incarnazione da 33 MHz, oltretutto, la scheda della GVP è di gran lunga la più veloce scheda acceleratrice che abbiamo provato sino a oggi. Il suo limite maggiore è il progetto della scheda di memoria, alla quale si può aggiungere memoria a 32 bit soltanto con costosi incrementi di quattro mega alla volta. La società produttrice insiste a dire che le RAM nibble-mode aumentano le prestazioni, ma i nostri test dimostrano che la scheda Impact non è più veloce della Hurricane 2800 (con la stessa frequenza di clock) anche se è abilitato il "burst-mode" di accesso alla memoria (con il burst-mode si può sfruttare pienamente la capacità delle RAM nibble-mode di accedere a quattro locazioni di memoria consecutive, leggendo la prima word da 32 bit in modo normale e le tre successive in un solo ciclo di clock a testa). Inoltre ci sembra che sarebbe preferibile un'interfaccia SCSI. Anche se il bus d'interfaccia AT costituisce un modo economico per aggiungere un hard disk, non dà però supporto a dispositivi come hard disk rimovibili, unità di backup a nastro e CD ROM. Malgrado queste riserve, comunque, la scheda della GVP è eccellente. Può essere configurata per adattarsi a esigenze di velocità e di prezzo estremamente varie, e anche senza la costosa espansione di memoria fornisce prestazioni che superano quelle degli acceleratori 68020 con memoria a 32 bit.

Per ulteriori informazioni contattare direttamente:

Commodore Italiana
(A2630 con 2 MB, L. 3.014.000 + IVA
A2630 con 4 MB, L. 4.066.000 + IVA)
Viale Fulvio Testi, 280
20126 Milano
(Tel. 02/661231)

Imtronics
(Hurricane 2800 con 2 MB, $1.195)
12301 S.W., 132 Court
Miami, FL 33186, USA
(Tel. 001/305/2559302)

Great Valley Products
(Impact A3000 con 4 MB, $2999
Impact A4000 con 4 MB, $3999)
PO Box 391
Malvern, PA 19355, USA
(Tel. 001/215/8899411) ■

NOVITÀ HARDWARE

IL NUOVO AMIGA 3000: SI APRE L'ERA DELLE PERSONAL WORKSTATION

di Morton A. Kevelson, Matthew Leeds, Massimiliano Lisa e Mauro Gaffo

A Manhattan, il 24 aprile, è nato ufficialmente il nuovo grande erede della famiglia Amiga: l'Amiga 3000. Oltre alla macchina vi presentiamo anche il WB 2.0, AmigaVision, l'ECS e il VDE

Meno di cinque anni fa, nell'agosto del 1985 per essere esatti, la Commodore presentò l'Amiga 1000 al Lincoln Center di New York. Quest'anno la Commodore ha scelto ancora New York per annunciare ufficialmente l'ultima creazione dei maghi di West Chester. La presentazione è avvenuta al Palladium, a Manhattan, nella Quattordicesima Strada. Benché il Palladium non sia elegante quanto il Lincoln Center (e, in effetti, i sobborghi di questo ex cinema vengono comunemente considerati una zona decaduta), da molti punti di vista la sala di quella che ora è probabilmente la più grande discoteca di New York si adatta bene a ciò che l'Amiga 3000 ha da offrire. Il nome dello spettacolo era "Multimedia Live!", e la sala principale del Palladium, con le sue due serie gemelle di 25 schermi, i suoi complessi di riflettori girevoli e il suo monumentale sistema di diffusione sonora (che può essere calato dal soffitto di 16 metri fin sopra le teste del pubblico) si è rivelata un'ottima scelta per ciò che l'Amiga 3000 rappresenta nel suo complesso.

Si è trattato della prima presentazione ufficiale e pubblica del nuovo nato, ma l'Amiga 3000 non era in realtà una sorpresa assoluta. Che nell'ambiente già si sapesse qualcosa lo testimoniano il numero di maggio di *Byte* e quello di giugno di *AmigaWorld*, già stampati in occasione della manifestazione. Entrambe le riviste hanno presentato ampi articoli sul nuovo computer Commodore, ma per far coincidere i tempi d'uscita con la presentazione ufficiale, la Commodore aveva fornito in anticipo le informazioni richieste. A Washington, prima dell'AmiEXPO che si è svolta a metà marzo, aveva anche messo alcuni esemplari della nuova macchina a disposizione di esponenti scelti della stampa.

Lo spettacolo è iniziato in sordina. Alcuni tizi dall'aria di tecnici con una cuffia sulla testa, in giacca nera sportiva, si affannavano qui e là tra controlli delle apparecchiature e prove di vario tipo. Agli orecchi della platea arrivavano di tanto in tanto brani di conversazioni improvvisate. Si è sentito un tecnico che diceva: «Questo

dovrebbe cacciar dentro qualche CPU nella porta seriale». La risposta è stata: «E la chiameresti una partenza a caldo, questa?». Per un po' mi è sembrato di trovarmi al lancio dello space shuttle, finché non mi sono ricordato che lo shuttle era già nello spazio da almeno due ore.

Una musica di sottofondo contribuiva a mantenere la suspense, e lo stesso scopo aveva un cronometro digitale (sui due grandi schermi televisivi all'estremità della sala) su cui scorrevano i centesimi di secondo del conto alla rovescia per dare inizio allo spettacolo. Il primo a parlare è stato nientedimeno che il professor Erwin Corey, il maestro di tutti i tempi delle dissertazioni tecnologiche, la cui immagine registrata è apparsa sullo schermo per spiegare che cos'è la multimedialità. Benché le spiegazioni del professore non abbiano chiarito granché, almeno ho saputo che la multimedialità non dev'essere sottovalutata: senza dubbio un consiglio di grande valore.

Completato il cerimoniale di apertura, i riflettori si sono puntati su Harry Copperman, il più recente esponente della lunga e onorata serie di presidenti della Commodore. Sotto la direzione dell'uomo di rame ("copper" significa "rame", *ndr*), la Commodore ha finalmente raggiunto un assetto definitivo per quanto riguarda il marketing dell'Amiga. Con l'Amiga 3000, intende sfondare nel mercato della multimedialità. Copperman ha affermato che l'Amiga era pronto per la multimedialità già da cinque anni, e che è stata la multimedialità a metterci tutto questo tempo per arrivare ai blocchi di partenza. Ora la multimedialità è finalmente arrivata e mentre gli altri computer si stanno affannando per raggiungerla, l'Amiga 3000 fa un ulteriore passo avanti. Per citare le parole di Lloyd Mahaffey, vice presidente marketing della Commodore statunitense: «Negli anni Settanta ci fu il word processing, negli anni Ottanta il desktop publishing... negli anni Novanta ci sarà la multimedialità».

Dopo la presentazione formale, il pubblico, oltre duecento inviati stampa ed esponenti di altri settori interessati all'avvenimento, ha potuto fare le sue domande al personale tecnico e provare personalmente l'Amiga 3000.

Girando qua e là al Palladium

Partecipavano alla manifestazione anche parecchi programmatori e produttori hardware, che avevano portato i loro prodotti e li esponevano sui tavoli disposti nella parte posteriore della sala. Andando da sinistra a destra, ecco chi era presente.

La **WordPerfect Corporation** sta rivalutando il suo interesse nei confronti dell'Amiga. Il futuro di *WordPerfect* verrà deciso sulla base dell'ultimo incontro avuto con la Commodore e sulla valutazione che la società darà del nuovo corso imboccato con la nascita dell'Amiga 3000. Se i risultati saranno positivi, si darà il via alla versione 5.0 di *WordPerfect*.

La **Precision Incorporated** ha portato una versione "pre-alfa... non toccate la tastiera altrimenti potrebbe andare in crash" del programma *Superbase 4* per Amiga. Si tratta di quella che fino a oggi era la versione Windows di *Superbase*, disponibile soltanto per i computer MS-DOS.

Il gruppo **Zuma** esponeva *TV*TEXT Professional* e *TV*Show*.

La **NewTek** era presente con l'atteso Video Toaster, e con una sorprendente demo del prodotto. A proposito di Video Toaster, sembra che il suo momento sia finalmente arrivato.

M.K.

L'AMIGA 3000 VISTO DA VICINO

Motorola 68030 a 25 MHz, 2 MB di RAM, ECS, VDE... Arriva la macchina multimediale per il professionista

Ma insomma, che cos'è l'Amiga 3000? Innanzitutto un computer basato sul microprocessore 68030 a 25 MHz. È equipaggiato con il coprocessore matematico 68882, e ha non meno di otto chip dedicati, due megabyte di RAM a 32 bit (un mega di fast RAM e uno di chip RAM; il computer può indirizzare direttamente 2 mega di quest'ultima e sulla scheda sono già presenti gli zoccoli per accogliere anche il secondo), 512K di ROM, un disk drive da 880K e un controller SCSI DMA ad alta velocità e a 32 bit, con un hard disk da 40 MB. Il prezzo di listino (IVA esclusa) di questa configurazione è stato fissato in 6.500.000 lire (negli USA $3999). Sono disponibili anche una versione con hard disk da 100 MB dal costo di 7.500.000 ($4499) e una versione più economica con l'accoppiata 68030/68881 a 16 MHz e hard disk da 40 MB a 5.500.000 lire ($3299).

Come si vede, i prezzi italiani sono più alti di quelli statunitensi. Probabilmente si è voluto sottolineare che il target dell'A3000 è prevalentemente quello professionale. Questa non è una macchina che sostituisce i modelli precedenti, ma è il completamento di una linea che propone l'A500 per il ragazzo, l'A2000 per l'hobbysta evoluto e l'A3000 per il professionista. Chi usa il computer soprattutto per giocare non ha bisogno di una macchina potente, veloce e con hard disk interno, e d'altra parte l'hobbysta evoluto o il piccolo professionista possono benissimo fare a meno di un'architettura a 32 bit, quando con l'A2000 possono avere ugualmente gli stessi modi video (si veda il box "Arrivano l'ECS e il VDE").

Un nuovo gradino dell'evoluzione?

La creazione dell'A3000, sia chiaro, non è un gigantesco passo in

avanti rispetto alla tecnologia Amiga sviluppata in passato. Alle sue spalle, però, è presente un notevole sforzo di progettazione, sia per quel che riguarda l'hardware, sia per il software, sia per il design. Il risultato è un computer che rappresenta l'evoluzione dei modelli precedenti: ne elimina le pecche e contemporaneamente se ne distacca dal punto di vista della velocità. È un'evoluzione nata per le applicazioni professionali.

Per darvi un esempio dell'accuratezza del generale lavoro di "revisione", potremmo citare i fastidiosi problemi di allineamento del disk drive al frontale presenti nell'A2000, che sono stati risolti con l'introduzione di un supporto di metallo al quale il drive si fissa con una sola vite. Per un uso monofonico, inoltre, non è più necessario un adattatore Y da collegare all'uscita audio, e collegare tastiera e mouse ai relativi connettori non è più scomodo come con l'A2000.

Le due porte mouse/joystick sono state spostate sul lato destro della macchina, così come il connettore della tastiera. Sul retro, l'Amiga 3000 ha le porte seriale, parallela, disk drive, SCSI, audio, RGB e VGA. Rispetto all'A2000, l'unità centrale è stata decisamente rimpicciolita, e si avvicina alle dimensioni dell'Amiga 1000. Si tratta di un design adatto allo stile delle ultime generazioni, che tendono a proporre elevata potenza all'interno di unità centrali dalle dimensioni ridotte.

L'ultimo nato non offre quindi un'architettura rivoluzionaria a 48 bit o una risoluzione di 3200 x 2560 pixel, bensì il meglio di quello che è oggi disponibile. Ovvero una nuova architettura a 32 bit, con registri, indirizzi e bus di dati da 32 bit. Si tratta di un passaggio importante rispetto ai 24 bit del precedente Zorro II, che consente alla macchina di far funzionare a pieno regime e senza limitazioni non solo il potente 68030, ma anche il nuovo chip 68040 – atteso per la fine dell'anno – che sarà in grado di eseguire 15-25 MIPS (milioni di istruzioni al secondo).

Rispetto ai modelli precedenti non mancano poi importanti punti di netta superiorità: niente più sfarfallio, nuova risoluzione da 1280 x 512 pixel e un AmigaDOS tanto migliorato da dover essere battezzato 2.0.

Per restare perfettamente al passo con gli ultimi sviluppi nel campo delle applicazioni, la macchina è inoltre corredata del nuovo sistema "autore" *AmigaVision*, che consente di trasformare il proprio computer in una stazione multimediale.

I nuovi chip custom

Il nuovo set di chip custom porta parecchi significativi miglioramenti al display dell'Amiga 3000. Tutti i vecchi modi video sono ancora disponibili, ma possono essere utilizzati anche senza interlace. Addio, vecchio sfarfallio dell'interlace! È comunque necessario un monitor VGA per sfruttare i nuovi modi video. A questo scopo, la Commodore offre un nuovo monitor a colori multisync 1950 per 815 mila lire. In aggiunta al modo non-interlace, il nuovo set di chip serve anche a raddoppiare la risoluzione di schermo orizzontale, mettendo a disposizione i modi 1280 x 256 pixel non interlace, oppure 1280 x 512 in interlace. Per questi modi video, i colori sono limitati a quattro su una palette di 64. C'è anche il "productivity mode", 640 x 480 pixel, anch'esso limitato a quattro colori.

I nuovi modi grafici superano praticamente ogni precedente imperfezione del display dell'Amiga. L'unico limite che rimane è quello della palette. La palette di lavoro, infatti, è sempre limitata a 4096 colori e i colori di schermo sono ancora limitati a 32. Per una vera stazione video, gli standard attuali richiederebbero una palette a 24 bit (16.777.216 colori) con 256 colori di schermo. Ufficiosamente, durante la presentazione negli Stati Uniti, si è saputo che la Commodore sta seguendo sia una linea a breve termine (la fine dell'anno), sia una linea a lungo termine, per superare la limitazione dei 4096 colori.

Oltre a Super Agnus, Super Denise (ECS) e Paula, sono presenti altri cinque chip dedicati. Due sono ver-

LE CARATTERISTICHE TECNICHE

- ❑ 68030/882 a 25 MHz oppure 68030/881 a 16 MHz
- ❑ Amiga Enhanced Chip Set (ECS)
- ❑ Video Display Enhancer (VDE)
- ❑ 1 MB di chip RAM espandibile internamente a 2 MB
- ❑ 1 MB di fast RAM espandibile internamente a 4 MB (con chip 256 x 4) o a 16 MB (con chip 1K x 4)
- ❑ Hard disk SCSI interno da 40 MB o da 100 MB (entrambi con tempo d'accesso di 19 ms)
- ❑ Orologio interno in tempo reale con batteria tampone
- ❑ 4 slot di espansione Zorro III (ognuno dei quali è compatibile con le schede dell'A2000)
  - 1 slot 100 pin
  - 2 slot 100 pin con estensioni stile AT
  - 1 slot 100 pin con estensione slot video stile A2000
- ❑ 1 connettore di espansione CPU da 200 pin
  - Possibilità di controllo della macchina da parte del processore Può supportare processori avanzati, RISC, CACHE
- ❑ Connettori esterni
  - 23 pin Amiga video (15,75 KHz)
  - 15 pin VGA (31,5 KHz)
  - Porta SCSI
  - Disk drive esterno/Seriale
  - Parallela/Audio stereo
  - Tastiera/2 Joystick
- ❑ Accesso a 32 bit alla ROM interna
- ❑ Accesso a 32 bit alla chip RAM
- ❑ Fast RAM a 32 bit, con controller dedicato, che supporta le DRAM "static column mode" e consente l'accesso sequenziale della CPU alla RAM
- ❑ Controller SCSI basato su DMA a 32 bit interna
- ❑ Unità centrale di dimensioni ridotte che può contenere fino a tre drive interni da 3,5" (usualmente 2 disk drive + 1 hard disk)
- ❑ Interruttore sulla parte anteriore
- ❑ Installazione di drive semplificata
- ❑ Supporto hardware e software sia per lo standard video statunitense a 50 Hz sia per quello europeo a 60 Hz

sioni potenziate di chip già presenti sull'A2000: Fat Gary, che fornisce la decodifica degli indirizzi ed è una sorta di collante tra le varie parti del sistema, e Fat Buster che si occupa del DMA e gestisce il bus di espansione adattandolo allo standard Zorro III. Il terzo chip, Ramsey, controlla la fast RAM, mentre il quarto, Super DMAC agisce come un controller DMA per l'interfaccia del bus SCSI. Con il quinto, Amber, torniamo al video: il chip fornisce la logica di controllo del video display enhancer (VDE). Si tratta del dispositivo che trasforma il video da 15 KHz in interlace in 31 KHz non-interlace (si veda il box "Arrivano l'ECS e il VDE").

La memoria

La RAM di sistema consiste in due megabyte in chip standard (di cui uno solo è installato al momento dell'acquisto). Sulla scheda madre possono essere inoltre installati fino a 16 megabyte addizionali di RAM. Per risparmiare spazio, questi chip possono essere chip zig-zag da 1 megabit (fino a un massimo di 4 megabyte), oppure ZIP da 4 megabit (fino a 16 megabyte). Entrambi i tipi di chip sono progettati per essere installati in posizione verticale, risparmiando spazio sulla scheda madre. Il sistema, di per sé, è in grado d'indirizzare fino a 1,7 gigabyte di RAM. Quando viene installata l'espansione di memoria, l'originale megabyte di fast RAM viene scollegato e reinstallato negli zoccoli di chip RAM vuoti presenti nella scheda madre. Questo permette di aumentare la chip RAM fino a due megabyte. C'è da aspettarsi che uno dei primi miglioramenti che verranno realizzati dai nuovi possessori dell'Amiga 3000 sarà l'installazione di fast RAM supplementare per un minimo di 4 megabyte, portando simultaneamente la memoria chip al suo massimo di due mega. Va rilevato che mentre il 68030 accede a tutta la RAM di sistema come memoria a 32 bit, invece i chip dedicati vi accedono come memoria a 16 bit.

Da sinistra a destra: la porta seriale, disk drive, SCSI, parallela, le uscite video RGB e VGA, il regolatore VDE, il selettore VDE, i canali audio destro e sinistro

LA MAPPA DI MEMORIA DELL'A3000

$80000000
Zorro III Area di espansione
$10000000
Memoria a 32 bit Area di espansione
$08000000
Area scheda madre A3000
$01000000
$00000000

$01000000
ROM su scheda madre
$00F00000
Zorro II I/O
$00E80000
Area dei registri della scheda madre dell'A2000
$00B80000
Zorro II I/O Area di espansione
$00A00000
Zorro II Area di espansione della memoria
$00200000
Memoria chip dell'Amiga
$00000000

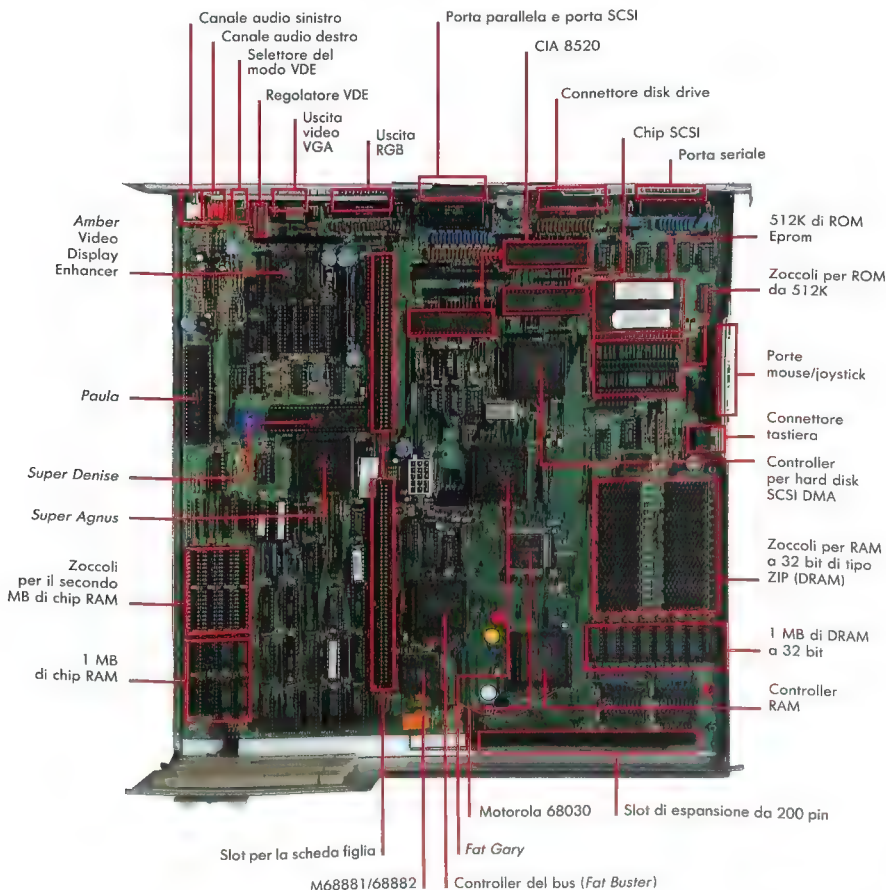
Una nota per i programmatori. La memoria inizia a $07FFFFFF, e cresce verso il basso fino a $07F00000 nel caso di un megabyte, o $07C00000 nel caso di 4 megabyte. Questo va chiaramente oltre lo spazio d'indirizzamento a 24 bit di A500 e A2000. Perché i programmi scritti per i modelli precedenti siano compatibili è quindi necessario che gli otto bit superiori d'indirizzamento non siano stati utilizzati per nient'altro che per l'indirizzamento.

Il motore della macchina

Osserviamo ora l'interno. La scheda madre è provvista di un nuovo slot CPU da 200 pin, progettato per schede ultra veloci di RAM statica, per prodotti CACHE, per nuovi chip come il 68040, per schede processore RISC...

Ci sono inoltre due slot di espansione ai quali è già collegata una "scheda figlia" multislot. La scheda figlia è installata verticalmente e le schede di espansione s'innestano orizzontalmente. La scheda figlia contiene

quattro slot d'espansione Zorro III in grado di accogliere schede da 16 bit (in stile A2000), oppure le nuove schede d'espansione a 32 bit. Gli slot da 100 pin acquisiscono capacità da 32 bit effettuando un'operazione di multiplex sulle linee dei dati. Dei quattro slot, uno è un normale 100 pin, due slot hanno le estensioni stile AT per l'installazione di una Bridge-board, l'ultimo ha invece come estensione uno slot video stile A2000.

Benché la scheda figlia dell'A3000 fornisca alla macchina una grande possibilità d'espansione, è meno flessibile di quanto non lo sia la sistemazione degli slot d'espansione dell'A2000. Gli utenti dovranno fare bene attenzione alle loro scelte nell'ampliamento del sistema. Per fortuna, l'A3000 raccoglie sulla scheda madre già la maggior parte dell'hardware che doveva essere aggiunto all'A2000. Nonostante ciò, è prevedibile che tra le prime periferiche prodotte da società esterne alla Commodore ci sarà una scheda-figlia alternativa. Un'altra scheda potrebbe combinare le funzioni di memoria supplementare e di genlock.

AMIGA 3000

Per chi non lo sapesse, il nome Zorro viene dal nome di una delle schede-prototipo dell'A1000. La scheda Zorro sostituì la precedente Lorraine e fu la scheda sulla quale vennero decise quasi tutte le specifiche d'espansione. La Zorro III dell'Amiga 3000 garantisce compatibilità con le periferiche disegnate per l'A2000, aggiungendo molte nuove caratteristiche. Per esempio è decisamente più veloce della Zorro II e permette di accedere a un'area d'indirizzamento che tocca gli 1,7 gigabyte.

Il sistema operativo 2.0

Anche nel sistema operativo ci sono notevoli miglioramenti. Il nuovo *Workbench* ha un eccellente aspetto tridimensionale, superiore a quello di qualunque altra interfaccia utente presente oggi sul mercato. I nuovi modi video senza interlace contribuiscono a migliorare ulteriormente l'aspetto del *Workbench*. Si può inoltre accedere dal *Workbench*, senza bisogno di icone, anche ai programmi che precedentemente dovevano essere mandati in esecuzione dal CLI. Va anche detto che il sistema operativo include requester standard che possono essere utilizzati dai programmatori. In effetti, il sistema operativo è stato migliorato a tal punto che la nuova release si chiama 2.0: la tanto chiacchierata versione 1.4, in pratica, non uscirà mai.

Tra le caratteristiche del *Workbench 2.0* troviamo cassetti più rapidi da aprire e più semplici da creare; un maggior numero di operazioni asincrone (si può per esempio caricare un'applicazione mentre si sta copiando un file); la possibilità d'interrompere una procedura mentre è ancora in esecuzione; uno Shell con più comandi e la possibilità di svolgere tutte le funzioni del CLI da *Workbench*.

Inoltre è stato incluso nel software sistema l'*AREXX*, sono supportati drive RAM multipli, il dispositivo TrackDisk è più veloce e il FastFileSystem è in ROM. Intuition fornisce maggior supporto ai gadget personalizzati, nuovi gadget stringa con opzioni potenziate, supporto diretto agli schermi overscan, scroll, autoscroll e schermi public. Ulteriori miglioramenti includono lo scaling delle bitmap, il supporto alle ColorFont e un rendering del testo più rapido. Infine va detto che dal *Workbench* si possono selezionare i display in PAL o in NTSC e si può richiedere l'attivazione del modo video Productivity (640 x 480).

Sopra: il sistema autore AmigaVision apre le porte alla multimedialità. Sotto: il nuovo Workbench versione 2.0

Per dimostrare che la Commodore fa veramente sul serio con la multimedialità, l'Amiga 3000 viene fornito insieme ad *AmigaVision* (si veda l'articolo "*AmigaVision*, anno zero"), il nuovo sistema autore della Commodore per le applicazioni multimediali. Dopo aver visto con i nostri occhi le sue potenzialità, possiamo tranquillamente affermare che questo pacchetto sarà in grado di soddisfare le esigenze anche degli utenti davvero "hyperattivi". *AmigaVision* sarà acquistabile anche dagli attuali utenti di A500/2000 a un prezzo che si aggira sulle 200 mila lire.

La versione 2.0 del sistema operativo e *AmigaVision* sono inclusi nella confezione dell'A3000. A partire dalla fine dell'anno saranno presenti anche in quella dell'A2000, mentre non ne è annunciata la presenza nelle scatole degli A500.

Il collegamento in rete

La Commodore dimostra di aver preso sul serio anche il collegamento in rete. Sono infatti stati annunciati una serie di prodotti destinati a far aumentare i potenziali utenti di un sistema Amiga.

L'adattatore per rete Ethernet (A2065, L. 590.000) funziona a 10 megabit al secondo e supporta protocolli ANSI 802.3 di tipo Ethernet, gestendo sia la versione tick sia quella thin.

L'adattatore per rete Arcnet (A2060, $229, disponibilità italiana n.c.) supporta protocolli Arcnet a 2.5 Mbps, gestendo fino a 256 nodi e permettendo l'autoboot.

L'AS220 Amiga Client per Novell NetWare ($149, disponibilità italiana n.c.) funziona con l'A2060 per comunicare con network Novell funzionanti con la versione 2.15 (o più recente) del Novell NetWare. Il software dispone di password e fornisce due interfacce utente: una basata sul testo e l'altra sull'uso del mouse. Il software TCP/IP e NFS ($199, disponibilità italiana n.c.) permette all'Amiga il collegamento con altre reti che supportano questi due standard, come, per citare un esempio, quella del ministero della Difesa statunitense.

continua a pagina 34

ARRIVANO L'ECS E IL VDE

Tutto quello che si deve sapere sui nuovi chip Commodore per Amiga 500/2000 e per il nuovo 3000

L'Amiga 1000 è nato con tre chip dedicati – Agnus, Denise e Paula – che svolgevano importanti compiti legati alle capacità audio/video della macchina. Agnus era già diventato Fat Agnus nei modelli A500 e A2000, e oggi di questi chip solo Paula rimane immutato. L'Enhanced Chip Set infatti sostituisce a Denise e a Fat Agnus le loro nuove versioni potenziate.

Super Agnus e la chip RAM

Il primo dei due chip di cui è costituito l'ECS è Super Agnus, una nuova versione del Fat Agnus di A500 e A2000 La novità più importante è costituita dal fatto che la massima quantità di memoria direttamente controllabile dai chip dedicati a grafica e sonoro passa da 512K a 1 MB.

Questo porta a una diversa suddivisione della memoria. Fino a oggi i chip dedicati potevano accedere direttamente solo ai primi 512K di memoria dell'A500 e dell'A2000, ovvero alla chip RAM Agli altri 512K dell'A2000 (e opzionali nell'A500) aveva invece accesso solo il microprocessore, e questo settore era chiamato fast RAM. Con l'introduzione di Super Agnus, i 512K di fast RAM diventano invece parte integrante della chip RAM e sono quindi accessibili ai chip dedicati. Sottolineamo che Super Agnus amplia la possibilità d'indirizzamento diretto a 1 megabyte nell'A500 e nell'A2000; nell'A3000, invece, questo valore sale a 2 megabyte.

Un altro potenziamento riguarda l'aumento delle dimensioni degli oggetti grafici che il Blitter (il coprocessore che si occupa degli spostamenti) è in grado di muovere: si passa da 1K x 1K a 32K x 32K

Super Agnus è anche in grado di modificare opportunamente la frequenza di scansione generando un display in PAL (lo standard televisivo europeo) oppure in NTSC (lo standard statunitense)

Quanto ai vantaggi pratici, diciamo subito che la maggior quantità di chip RAM indirizzabile può migliorare le prestazioni dei singoli programmi grafici, e anche permettere l'uso di più programmi contemporaneamente. Le dimensioni di 32K x 32K utilizzabili dal Blitter rendono più semplice per i programmatori la manipolazione e lo scorrimento di grandi schermi virtuali. Infine, la possibilità di cambiare lo standard video rende più semplice lo scambio di software tra Europa e Stati Uniti

Super Denise e la risoluzione

Innanzitutto una precisazione utile per chi, non possedendo l'ECS, decidesse di espandere la sua macchina solo con Super Denise· mentre Super Agnus può funzionare anche senza Super Denise, Super Denise richiede assolutamente la presenza di Super Agnus

Super Denise offre due nuove possibilità. Prima di tutto permette di generare, con il normale monitor 1084, un numero di pixel per riga doppio rispetto a quello di un normale display in alta risoluzione. In pratica ogni pixel diventa largo la metà del normale, questo nuovo modo video prende il nome di **Super Hi-Res** e offre una risoluzione non-interlace di 1280 x 256 pixel e di 1280 x 512 pixel in interlace. Il doppio di pixel vuol dire però la metà di colori, e quindi sono utilizzabili al massimo quattro colori scelti da una palette di 64. Con questo modo video, inoltre, non funziona il multitasking

In secondo luogo consente al software di controllare i segnali di sincronizzazione e di vertical blanking. Questo si traduce nella possibilità di alterare il segnale video per utilizzare qualunque tipo di monitor. Una possibile applicazione è il modo video **doublescan**, nel quale la velocità di scansione orizzontale viene raddoppiata per poter utilizzare i monitor VGA. Il doublescan dimezza la risoluzione orizzontale e raddoppia quella verticale (doppio delle righe con la metà di pixel per ogni riga) Modificare la velocità di scansione non modifica il numero di colori utilizzabili, ma richiede un monitor diverso dal 1084, un monitor in grado di accettare velocità di scansione più elevate, ovvero in standard VGA o multiscan

Il doublescan rivela la sua utilità quando viene impiegato in combinazione con il nuovo modo video Super Hi-Res, dando origine al modo **Productivity** che genera un'immagine da 640 x 480 pixel. Un display Super Hi-Res, come abbiamo già visto, può utilizzare solo quattro colori (due bitplane) ed essendo multiscan ha bisogno di un monitor VGA

In sostanza Super Denise offre quattro nuove risoluzioni Productivity (640 x 480), Productivity interlace (640 x 960), Super Hi-Res (1280 x 256), Super Hi-Res interlace (1280 x 512). Grazie all'interessante possibilità d'intervenire sul segnale di sincronizzazione è comunque possibile dar vita ad altre risoluzioni, il che potrebbe rivelarsi utile per particolari programmi

Per gli amanti delle applicazioni video, Super Denise porta con sé altre sorprese utilizzabili via software l'opzione ChromaKey, che permette a qualunque registro di colore di dar vita al trasparente, l'opzione BitPlaneKey che permette di rendere un bitplane trasparente, l'opzione BorderNotTrasparent, che permette di rendere opaco il bordo anche nel caso in cui il suo colore sia trasparente, l'opzione BorderBlanking, che rende invece il bordo trasparente anche se il colore dello sfondo è stato reso opaco

Oltre l'ECS: il VDE

Come abbiamo visto, l'ECS di per sé non risolve il problema del flickering (sfarfallio) in interlace né per l'A500/2000 né per il 3000. Per ottenere un effetto di de-interlace sui normali modi grafici dell'Amiga e avere lo

stesso numero di colori dei modi standard (fino a 4096) è necessario un apparecchio come il FlickerFixer della MicroWay. Nell'Amiga 3000 un flicker fixer è stato integrato sulla scheda madre, ed è costituito dal chip Amber, chiamato anche Video Display Enhancer (VDE). Per gli utenti dell'A2000 la Commodore ha previsto invece la scheda A2320 (L. 460 000), mentre per l'A500 non è previsto (sempre da parte della Commodore) nessun VDE Il FlickerFixer della MicroWay, la scheda A2320 e il chip Amber dell'A3000 richiedono tutti l'impiego di un monitor multisync o VGA. Per poter utilizzare modi video vecchi e nuovi, gli utenti hanno comunque bisogno di un monitor VGA multisync, in grado cioè di visualizzare i 31 KHz del display enhancer e i 15 KHz degli altri modi video.

Per quel che riguarda l'A3000, il VDE effettua un'operazione di de-interlace sul segnale interlace a 15 KHz trasformandolo in un display non-interlace a 31 KHz. Il VDE contiene una logica che gli permette di non effettuare il de-interlace sui display non-interlace, ma di effettuare invece una doppia scansione. Sul retro della macchina è presente un interruttore che consente di disattivare il VDE, il che si può rivelare utile per esempio durante l'uso di occhiali per applicazioni in 3-D.

Cosa possono fare gli utenti di A500/1000/2000

I nuovi A2000 montano ormai Super Agnus da diversi mesi (e da fine anno monteranno anche Super Denise). Per essere certi che la macchina contenga il fatidico chip, è sufficiente impartire il comando AVAIL da CLI; se Super Agnus è presente, il terzo elemento nella prima riga è 1040152 (altrimenti il valore è dimezzato).

Anche gli A500 hanno Super Agnus di serie (e non è invece prevista la futura installazione di Super Denise) Per verificarne la presenza è però necessario aprire il computer. se il nuovo chip è presente, sulla scheda madre devono esserci 4 chip di RAM anziché 16. Dal momento che l'A500 ha una dotazione di soli 512K di memoria, il computer è impostato per vedere una chip RAM da 512K; e questo anche se s'installa l'espansione da 512K! Per poter disporre della chip RAM raddoppiata è necessario operare anche su due jumper della scheda madre, per cui si è obbligati a ricorrere a un centro di assistenza.

Chi non dispone di Super Agnus può richiederne l'installazione in un centro di assistenza (il tutto dovrebbe costare circa 150 mila lire).

Per quel che riguarda Super Denise (che sarà disponibile da settembre in avanti) gli utenti di A500/2000 devono rimuovere il chip vecchio dallo zoccolo e installare al suo posto quello nuovo.

Un discorso a parte è invece necessario per gli A1000. Mentre per l'installazione di Super Denise è sufficiente estrarre il vecchio chip Denise e rimpiazzarlo con quello nuovo, per quel che riguarda Agnus le cose cambiano Mentre Fat Agnus (di A500 e A2000) e Super Agnus hanno dimensioni uguali, Agnus e Super Agnus sono fisicamente diversi e non è possibile montare quest'ultimo sullo zoccolo del primo. Ufficialmente l'A1000 non è più supportato dalla Commodore e quindi chi vuole rimanere aggiornato non può far altro che comprare i nuovi modelli (negli Stati Uniti è stata avviata una favorevole campagna di trade-in).

Ma ecco che arriva la soluzione. Negli USA (ma siamo sicuri che li vedremo presto anche in Italia) sono infatti disponibili Rejuvenator ($499.95) della Expert Services (5912 Centennial Circle, Florence, KY 41042, USA, tel. 001/606/3719690) e DVS Wonder ($399 95) della Delaware Valley Software (PO Box 2007, Upper Darby, PA 19082-0507, USA, tel 001/215/4469227). Si tratta di due schede che sostituiscono completamente la scheda figlia presente sull'A1000 aggiungendo 1 MB di chip RAM e lo zoccolo per ricevere Super Agnus

Per quel che riguarda il VDE, gli utenti di A2000 che lo desiderano possono acquistare la scheda Commodore A2320 a 460 000 lire più IVA. Per gli A500/1000 la Commodore non fornisce nessuna possibilità di eliminare il flickering Ma ecco che dagli USA arriva la soluzione. per chi fosse interessato, è stato annunciato da un produttore indipendente Flickoff per A500/1000/2000 ($399, M.A.S.T., 1395 Greg. St. #106, Sparks, NV 89431, USA, Tel. 001/702/3590444). Vi daremo ulteriori informazioni nei prossimi numeri di *Commodore Gazette*. **M.L.**

Così sentenziò la redazione

Ci troviamo dinanzi al completamento di una famiglia di computer con l'aggiunta di un prodotto che si indirizza all'utenza professionale. Ovviamente la macchina non è disegnata con l'intento di sostituire l'MS-DOS negli uffici per quel che riguarda word processing, database o contabilità. Ha invece l'etichetta di computer "creativo", destinato a compiti quali la postproduzione video, l'ausilio ai musicisti (sequencer, notazione, campionamento...), il CAD, il desktop publishing, la computergrafica, le simulazioni d'ambiente in 3D... E poi non bisogna dimenticare l'importante prospettiva della multimedialità, alla quale l'A3000 è decisamente portato e che apre orizzonti come didattica, presentazioni dinamiche, simulazioni d'ingegneria, produzione video...

Il grande merito della Commodore è quello di essere riuscita a dotare il suo nuovo computer di caratteristiche che i concorrenti diretti mettono in vendita a un prezzo superiore (la considerazione vale anche per il mercato italiano, ma soprattutto per quello USA). La Commodore ha realizzato l'hardware e il software sistema per far decollare l'Amiga nell'ambito della "professionalità creativa". Ora resta soltanto da vedere quali saranno le reazioni del mondo.

Per la linea Amiga si tratta di un passo avanti, di un nuovo gradino evolutivo, anche se come abbiamo detto il passo non è gigantesco. In ogni caso dal gradino raggiunto possiamo dire che si comincia a intravedere una nuova era: se gli anni Ottanta sono stati gli anni dei PC (Personal Computer), gli anni Novanta saranno quelli delle PW (Personal Workstation) e l'Amiga 3000 ne è senza dubbio uno dei capostipiti.

M.L. e M.K.

AMIGAVISION, ANNO ZERO

Passata l'era del word processing e del desktop publishing, immagini, suoni e animazioni danno vita alla multimedialità

Nei circa dieci anni passati da quando i computer sono entrati prepotentemente nel mercato "casalingo", la creazione del software è sempre stata qualcosa di misterioso per gli utenti. Anche se al giorno d'oggi ogni nuovo sistema operativo nasce già provvisto di un adeguato linguaggio di programmazione, sembra proprio che la prospettiva d'imparare a crearsi i programmi da soli sia qualcosa che intimidisce la maggior parte degli utenti. I semplici programmi dimostrativi sono facili da capire, ma i concetti che si celano dietro alle funzioni di controllo di grafica, sonoro, animazione, database e interfacciamento con le periferiche, sono evidentemente troppo complessi per l'utente medio.

Negli ultimi dieci anni, però, i computer sono via via diventati sempre più facili da usare. Il salto dalla linea-comando di sistemi MS-DOS alle interfacce grafiche di sistemi come il *Workbench* ha offerto alla nuova generazione di utenti una "curva di apprendimento" molto più dolce per quanto riguarda l'impiego del sistema operativo. Ora la Commodore ha voluto portare la stessa "facilità d'uso" anche nel settore della programmazione, e a questo scopo ha creato *AmigaVision*.

Multimedia!

AmigaVision è un sistema di programmazione interattivo a icone per la creazione di presentazioni "multimediali". Ed ecco che spunta una parola un po' misteriosa, che da qualche tempo si sente sempre più spesso: *multimedia*. Si tratta di una parola non facile da definire, e che ha dato origine a qualche confusione tra gli utenti. Prima di proseguire, quindi, cerchiamo di definirla nel modo più chiaro possibile.

Per spazzare via ogni dubbio, diciamo subito che un programma che crea un'immagine, che impagina un testo, che suona un pezzo musicale, *non è* un programma multimediale. Ci avviciniamo un po' al concetto di multimedialità se pensiamo alle applicazioni in grado di mostrare un'animazione mentre contemporaneamente suonano un brano musicale. In realtà il multimedia è un sistema d'integrazione che consente all'utente finale di "fondere" e manipolare in vari modi disegni, animazioni, suoni, testi e immagini video, utilizzando sempre la stessa interfaccia.

A questo punto appare evidente che è improprio parlare di *programma* multimediale: la multimedialità è una tecnologia, che nasce allo scopo di controllare contemporaneamente diversi mezzi elettronici come lettori di dischi audio e video, monitor di computer, sintetizzatori, disk drive e via dicendo. Grazie a questa tecnologia si possono creare – con semplicità e immediatezza – elaborate presentazioni, corsi scolastici, pubblicità, video musicali e qualunque opera di "integrazione" la nostra fantasia ci suggerisca. I singoli elementi del prodotto finale sono immagini e suoni digitalizzati o creati con altri programmi, testi scritti con normali word processor, animazioni "importate" e così via. La tecnologia multimediale rende possibile fondere il tutto in un prodotto unico, legando le diverse fasi della presentazione con vari effetti speciali.

AmigaVision, in particolare, permette di organizzare uno "schema visivo" (basato su icone) del flusso logico della propria presentazione. Il materiale audiovisivo viene preparato in anticipo con i consueti programmi di disegno, musica, animazione, scrittura, e poi si crea la sequenza finale spostando le icone sull'apposita griglia visualizzata sullo schermo. Alcune icone rappresentano sequenze o immagini video, disegni, animazioni, suoni, testi; altre rappresentano gli elementi di controllo (cioè i vari "if...") della presentazione. *AmigaVision* sfrutta lo schema così creato (e alcune informazioni supplementari fornite sempre dall'utente) per costruire e mandare in esecuzione la presentazione.

Il diagramma di flusso

AmigaVision accetta input provenienti da mouse, tastiera e schermo tattile. Per il mouse e lo schermo tattile si possono definire gli "hit box", ovvero settori di schermo a cui viene assegnato un certo significato. Quando l'utente porta il cursore del mouse all'interno dell'hit box e preme il pulsante, viene svolta una certa azione. Questo trucco è utile per le selezioni dei menu, per gli help e così via. Per quanto riguarda la tastiera, tutti i tasti alfanumerici possono essere considerati come input validi da parte dell'utente, e si possono assegnare particolari significati ai tasti funzione.

Il diagramma di flusso delle applicazioni *AmigaVision* può contenere anche "strutture di controllo", ovvero cicli e salti condizionati, che rendono possibile una struttura più elaborata che non il semplice flusso sequenziale dei diversi elementi. Le strutture di controllo disponibili comprendono: if-then, if-then-else, branch (scelta di uno dei possibili percorsi nel flusso), salti non condizionati, loop, chiamate di sotto-routine. Tutte queste strutture di controllo sono create e organizzate tramite icone.

Proprio come nei normali linguaggi di programmazione, gli input dell'utente e le strutture di controllo hanno bisogno di variabili ed espressioni per raccogliere e immagazzinare i dati, e per valutare le condizioni. Per creare variabili ed espressioni per le valutazioni condizionali, si usa un "editor di espressioni" che mette a disposizione operatori matematici, funzioni, simboli e operatori logici.

C'è poi l'"editor di oggetti", per creare sullo schermo elementi di vario tipo (per esempio rettangoli e cerchi), che possono servire anche come hit box per gli input tramite mouse o schermo tattile. L'editor di oggetti consente anche di classificare come hit box una semplice sequenza di testo, fornendo persino una scelta di fonti e di colori.

Il controller per videodischi permette d'includere nella presentazione sequenze video o singoli fotogrammi, includendo o escludendo l'audio. Per le sequenze basta indicare il fotogramma iniziale e quello finale.

Un'altra possibilità è l'esecuzione di una sequenza musicale tramite un'opportuna strumentazione esterna: basta selezionare l'icona MIDI per inviare l'output a una batteria elettronica o a un sintetizzatore collegati all'elaboratore.

Per creare e manipolare dati è stato creato anche un "editor di database", che permette di creare un database di tipo standard, ovvero con la possibilità di aggiungere, aggiornare e cancellare singole voci e interi file. Le informazioni contenute nel database possono provenire da dati inseriti direttamente dal programmatore, oppure dalle risposte fornite dagli utenti (come per esempio le risposte di una classe di studenti a un test).

Sui blocchi di partenza

Per cominciare, cerchiamo di stabilire quale potrebbe essere la configurazione di lavoro ottimale. *AmigaVision* richiede un minimo di 1 MB di memoria, e questa esigenza non si può aggirare. Inoltre, benché sia possibile realizzare anche applicazioni eseguibili da floppy disk, è senz'altro consigliabile avere a disposizione un hard disk.

Poiché il sistema dà supporto anche a un lettore di dischi laser, per alcune applicazioni sarà senz'altro necessaria una periferica di questo tipo. Attualmente, i lettori di dischi laser gestibili dal sistema sono: Philips 405, 410 e 835; Pioneer 4200; Sony 1200, 1500, 1550 e 2000. Viene accettato anche output prodotto dal lettore di nastri Sony Umatic9. Se si utilizzano periferiche video può essere utile anche un genlock, ovvero un dispositivo che permette di sovrapporre le immagini dell'Amiga a quelle di una fonte video.

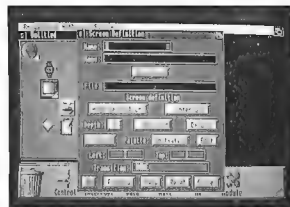
Attrezzati di tutto punto, possiamo ora esaminare in maggior dettaglio il programma. Lo schermo di apertura di *AmigaVision* contiene una griglia (la finestra di flusso, o Flow Window) su cui appare, in alto a sinistra, un'icona Module. Nella parte bassa è presente il menu iconico principale, con l'icona Trashcan a sinistra. Sul menu principale di *AmigaVision* ci sono sei icone, ciascuna delle quali rappresenta un sotto-menu (se si preme il pulsante del mouse su una delle icone principali, compare una nuova serie di icone). Per collocare un'icona sulla griglia, basta portarvi sopra il cursore, premere il pulsante del mouse e "trascinarla" nella posizione desiderata. Per eliminare un'icona basta spostarla nel cestino con lo stesso sistema.

Ogni icona dev'essere inizializzata: dopo averla portata sulla griglia, si fa comparire il suo "requester di definizione" premendo due volte il pulsante del mouse. Esaminiamo per esempio il requester dell'icona Screen (schermo): sono presenti un campo "nome" e un campo "memo"; un "pulsante directory", che fa comparire un requester di file tramite il quale si seleziona l'immagine da usare per lo schermo; un campo "filename"; alcuni "pulsanti di schermo" che servono per impostare la risoluzione, l'overscan, l'interlace e la palette; un toggle per il cursore; controlli di posizione...

C'è anche un'opzione di prova che serve per valutare subito la qualità dell'immagine e della transizione. Tutte le opzioni citate possono essere impostate tramite mouse, tranne le voci che richiedono input numerici o di testo.

L'organizzazione del programma

Tutte le icone collocate sulla griglia sono in relazione con quelle che già vi si trovano. In *AmigaVision* ci sono quattro tipi di relazioni tra le icone: sibling (sorella), parent (genitore), child (figlia) o partner (amica). Un'icona collocata sotto un'altra ne diventa la sorella. Le icone sorelle vengono eseguite nell'ordine in cui vengono incontrate. Le icone genitore hanno un piccolo triangolo chiaro nell'angolino in basso a destra. Quando una nuova icona viene associata a un'icona genitore, il triangolo diventa pieno e l'icona associata diventa icona figlia.

Quando un'applicazione *AmigaVision* va in esecuzione, un selettore interno esegue i comandi implicitamente definiti nell'icona genitore, poi passa alla prima figlia a sua volta seguita da tutte le sorelle, e così via finché l'intero flusso dell'applicazione non è stato esaurito. Dopo le figlie, vanno in esecuzione le successive sorelle del genitore.

Alcune icone di controllo (if-then, if-then-else) hanno la necessità di un'icona "amica" al loro fianco. Un'icona diventa "amica" di un'altra quando viene collocata al suo fianco, sulla destra. Viene eseguita per prima l'amica di sinistra, e poi quella di destra. Altre icone di controllo (come Call o Goto) richiedono un'amica che faccia riferimento a un'icona collocata in un altro punto della griglia.

Come già abbiamo accennato, *AmigaVision* contiene un editor di oggetti, grazie al quale si possono creare sullo schermo rettangoli, cerchi, linee, ellissi, poligoni e testi. Si possono creare campi per i dati e finestre per i testi. Ogni elemento creato con l'editor di oggetti può essere utilizza-

continua a pagina 38

1, 2, 3000!

Incontro stampa con la Commodore Italiana

Il 24 aprile 1990, in contemporanea in tutto il mondo, le varie filiali nazionali della Commodore hanno annunciato la presentazione del nuovo Amiga 3000, la macchina multimediale che - secondo le intenzioni della casa madre - dovrebbe lanciare finalmente l'Amiga nel mercato professionale. Negli Stati Uniti la presentazione è stata un vero e proprio show, e anche alcuni Paesi europei (come la Francia) hanno seguito l'esempio degli americani. In Italia, invece, non essendoci stato alcun avvenimento di rilievo a cui abbinare la presentazione, si è preferito puntare sui comunicati stampa e su una serie di incontri ristretti con i redattori dei periodici specializzati e con alcuni giornalisti ed esponenti del mondo industriale/commerciale.

Alla presenza dell'amministratore delegato della Commodore Italiana, Werter Mambelli, e di alcuni suoi collaboratori abbiamo quindi finalmente potuto ammirare l'Amiga 3000 "dal vivo", e scambiare quattro chiacchiere su tutte le novità previste nei prossimi mesi per gli utenti italiani.

Una Commodore che cambia

Cominciamo parlando un po' dell'evoluzione della CBM, che da qualche tempo ha imboccato un deciso cambiamento di rotta. Negli Stati Uniti, come già saprete, negli organismi direttivi della Commodore hanno fatto il loro ingresso personaggi come Bushnell, Copperman e alcuni altri nuovi dirigenti provenienti da realtà societarie di primissimo piano (Atari, Apple, IBM...). E i risultati della riorganizzazione cominciano a vedersi: la Commodore sta recuperando il terreno perduto negli anni scorsi.

In Italia i cambiamenti hanno preso forma nell'arrivo di Werter Mambelli (ex Compaq), e otto mesi della sua gestione già cominciano a dare i loro frutti. Lavorando con un organico di 40 persone, il nuovo amministratore è riuscito a chiudere il 1989 con 75 miliardi di fatturato (quarto posto nel mondo tra le varie sezioni nazionali), e per il 1990 si sta avviando trionfalmente verso la soglia dei cento miliardi Ma quello che più conta è l'atteggiamento di fattivo impegno e di creatività imprenditoriale che ha dato origine tra l'altro alla realizzazione di una struttura come la "Commodore Networking Division", la divisione che si occuperà delle reti Amiga.

Mambelli assicura inoltre che sono allo studio anche nuove iniziative per sfondare in un settore difficile ma promettente come quello scolastico.

Quanto alla struttura interna, si è dato il via a una grossa operazione di rinnovamento: la Commodore d'ora in poi si divide in due sezioni indipendenti, la divisione *consumer* (macchine a 8 bit e Amiga 500) e la divisione sistemi (PC40 e superiori, A2000/3000). In realtà l'Amiga 2000 viene considerato, sulla base di un'analisi di mercato, quasi a cavallo tra questi due mondi, lo sforzo, tuttavia, è quello di proporlo come una macchina che si stacca dal puro impiego ludico Invece non si può dire lo stesso dell'Amiga 500, irrimediabilmente legato al mercato del gioco, e per questa macchina non sono previsti (in Italia) aggiornamenti come l ECS, il *Workbench 2.0* e l'*AmigaVision*. Nello spot televisivo della Commodore, lo slogan che dovrebbe rilanciare il 500 dice più o meno· "Amiga 500 è pura immaginazione", e il *target* è quello dei ragazzi tra i dieci e i vent'anni. Come volevasi dimostrare

Il fatto importante, comunque, è che due realtà abbastanza indipendenti come il mercato professionale/amatoriale-evoluto e quello legato al puro divertimento, sono finalmente gestite in modo differenziato Questo renderà probabilmente possibile la creazione di un'assistenza tecnica più seria (non solo per il mercato professionale) e di una rete distributiva all'altezza del nome Commodore. L'altra grande novità "organizzativa", infatti, è la creazione di una rete di concessionari autorizzati che si occuperanno esclusivamente delle macchine professionali Attenzione: non si tratta di Commodore Point "riciclati", ma di punti vendita specializzati del tutto nuovi (anche se vi sono alcune eccezioni, nel caso di Commodore Point che abbiano dimostrato una professionalità e una competenza superiore alla media) Rimarranno, naturalmente, alcuni grossi punti vendita "tradizionali" che disporranno di tutta la gamma di prodotti Commodore. La rete di concessionari conta attualmente 35 nomi, alcuni del tutto nuovi per i prodotti Commodore, ed entro la fine dell'anno si prevede di arrivare a quota 80.

Il 3000 in Italia

Be', a questo punto avrete intuito che se ci sono i rivenditori dev'esserci anche la macchina. E infatti, nonostante i timori più o meno velati di ritardi, l'Amiga 3000 è già disponibile in Italia. Per ora ne è stata distribuita una quantità limitata, ma già da settembre i nuovi arrivi dovrebbero soddisfare ampiamente le richieste del mercato. Per la fine dell'anno, inoltre, anche tutti i nuovi 2000 saranno dotati - di serie - dell'ECS, del nuovo *Workbench 2.0* e del tanto chiacchierato sistema multimediale *AmigaVision*

Stando alle notizie attuali, i possessori del 500 o del 1000 interessati ad approfittare delle novità possono soltanto ricorrere a faticose acrobazie tecnologiche per espandere la macchina, e devono procurarsi e montare i nuovi chip per conto proprio. L'*AmigaVision* si troverà in vendita, ma i possessori dell'Amiga 500 ricordino che per utilizzarlo è necessario avere almeno 1 megabyte di RAM.

Tra parentesi, lasciateci fare un'osservazione "emotiva" l'Amiga 3000 è una macchina molto, molto attraente Esaminata anche solo sommariamente, sia all'esterno che all'interno, è più elegante sia del 2000 che del 500. È piccola, quasi esile, e rende evidente anche a una prima occhiata l'altissimo grado d'integrazione raggiunto. Tutte le schede d'espansione si inseriscono orizzontalmente, "rubando" meno spazio possibile Questi saranno forse particolari secondari, se li confrontiamo all'aumento della velocità operativa, all'eliminazione dello sfarfallio, all'efficienza del nuovo *Workbench*, ma, insomma .. anche l'occhio vuole la sua parte

Si è detto che l'Amiga 3000 è la macchina che sfonderà nel mercato professionale... e che la multimedialità dominerà il prossimo decennio. È vero, le aspirazioni sono queste, ma la stessa Commodore non si nasconde che il mercato *corporate*, quello delle grosse aziende, è ormai terreno quasi esclusivo del mondo MS-DOS A chi si rivolge quindi l'A3000? Soprattutto alle aziende, di qualunque dimensione, che abbiano la necessità di un eccellente computer grafico a un prezzo particolarmente aggressivo Un computer che possa occuparsi con efficienza di post-produzione video, presentazioni pubblicitarie, desktop publishing, animazioni, produzione di diapositive, CAD, desktop video informazioni sul punto vendita, e chissà quante altre applicazioni nei più disparati settori

Concludiamo riportando un'affermazione dell'amministratore delegato della Commodore Italiana «Ora come ora, l'Amiga 3000 è l'unica *vera* macchina multimediale sul mercato E non c'è alcun dubbio che, nella sua fascia di prezzi, resterà l'unica ancora per parecchio tempo»

M.G.

to come hit box, e può essere sovrapposto a uno schermo che contiene immagini di sfondo o immagini video. Questo permette di aggiungere interattivamente nuovi elementi all'applicazione.

Quando uno di questi oggetti "aggiunti" viene selezionato dall'utente, il programma può manifestare una reazione; per esempio l'oggetto può lampeggiare, oppure il computer può emettere un suono, o mandare in esecuzione un qualunque file musicale.

Il controller per videodischi fornisce un completo controllo sul lettore di videodischi collegato. Permette di visualizzare un singolo fotogramma o un'intera sequenza, e controlla perfettamente anche il sonoro: i canali audio di destra e di sinistra possono essere utilizzati indipendentemente l'uno dall'altro, oppure in contemporanea se il brano è stereo. *AmigaVision* può controllare fino a cinque dispositivi separati, con un Amiga attrezzato con diverse porte seriali. Miscelando la gestione dei videodischi con le capacità di database, non è difficile riuscire a crearsi un proprio programma che svolge funzioni di videoediting.

L'editor di database di *AmigaVision* mette a disposizione campi di testo lunghi fino a 255 caratteri, campi numerici di 19 cifre, valori booleani e campi dati. Ogni file di database può contenere fino a 128 campi con 4 mila caratteri ciascuno. È anche possibile definire alcuni "campi chiave" per il recupero dei dati. Nella creazione di un'applicazione *AmigaVision* si può intervenire sul database tramite menu a tendina e pulsanti. Tramite le icone del database, i dati che vi vengono immagazzinati possono essere selezionati, letti, scritti e inviati in uscita, sempre operando dall'interno dell'applicazione.

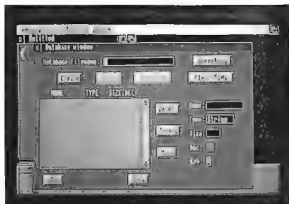
AmigaVision garantisce anche il supporto ad *AREXX*. Si accede ad *AREXX* tramite l'icona Execute, che propone due scelte: mandare in esecuzione un programma esterno, o lanciare file-script *AREXX*.

Prima di ricorrere ad *AREXX*, tuttavia, è necessario definire le variabili che saranno usate dai suoi file-script. A questo scopo, *AmigaVision* mette a disposizione due comandi *AREXX*: SETVAR e GETVAR, grazie ai quali si possono dichiarare i valori delle variabili che vengono definite.

Il futuro di AmigaVision

Il sistema *AmigaVision* è incluso nella confezione di tutti gli Amiga 3000, e si prevede che dalla fine dell'anno farà parte della dotazione di serie anche per gli A2000.

Più che la programmazione in se stessa, sostiene la Commodore, agli utenti interessa essere aggiornati su tutte le caratteristiche e le potenzialità che rendono l'Amiga un computer unico. E non c'è da stupirsene: che senso ha usare un linguaggio di programmazione testuale con un computer grafico?

Benché la Commodore non dia l'*AmigaVision* in omaggio (a meno che non si acquisti una macchina già equipaggiata con il nuovo sistema) cerca tuttavia di promuoverne la massima diffusione, un po' come fa la Apple per *Hypercard*, ed essenzialmente per le stesse ragioni. L'*AmigaVision* può ispirare la creazione di centinaia di applicazioni didattiche da parte di programmatori, sia dilettanti che professionisti: una base di prodotti che darà una notevole spinta all'ingresso dell'Amiga nelle istituzioni scolastiche. I docenti, inoltre, avranno la possibilità di realizzare programmi didattici più coinvolgenti e attraenti di quelli attualmente in circolazione, e a un costo molto più contenuto. L'hardware dell'Amiga è

decisamente meno costoso di quello della Apple o della IBM, e anche il software non raggiunge i loro livelli di prezzo.

C'è un'ultima osservazione da fare, che ci rende piuttosto ottimisti sul futuro di questo sistema. L'inclusione del supporto per dispositivi video esterni e per l'*AREXX* ci convince che *AmigaVision* non resterà un'applicazione isolata. La sua espandibilità incoraggerà gli utenti più intraprendenti ad aggiungere nuove funzioni, come accade con gli XCMD per *Hypercard*.

La Commodore ha dimostrato di guardare lontano, ideando e sviluppando un prodotto che rappresenta l'assicurazione che i limiti dell'Amiga sono ancora molto lontani da raggiungere.

M.L. e M.G.

LISTINO PREZZI DELLE NOVITÀ (IVA 19% ESCLUSA)

Prodotto	Prezzo
Amiga 3000 (16 MHz/40 MB)	L. 5.500.000
Amiga 3000 (25 MHz/40 MB)	L. 6.500.000
Amiga 3000 (25 MHz/100 MB)	L. 7.500.000
1930 (Monitor a colori VGA)	L. 720.000
1950 (Monitor a colori Multisync)	L. 815.000
A2024 (Monitor a fosfori bianchi)	L. 922.000
A2232 (Scheda per l'aggiunta di 7 porte seriali)	L. 390.000
A2320 (Scheda "video display enhancer")	L. 460 000
A2065 (Scheda Ethernet)	L. 590.000
A10 (Altoparlanti stereo)*	L. 60.000 (la coppia)

* necessari perché alcuni modelli di monitor non hanno altoparlanti

MODI VIDEO "DINAMICI" CON DIGIVIEW 4.0

Arriva per l'Amiga DigiView Gold 4.0, uno dei più importanti pacchetti per l'elaborazione video oggi disponibili sul mercato internazionale

di Joel Hagen

Fin dalla sua prima apparizione, il videodigitalizzatore della NewTek si è distinto per le sue eccezionali capacità ed è diventato, grazie anche al costo relativamente contenuto, uno dei prodotti più diffusi nella comunità degli utenti Amiga. Ma per mantenerlo al passo con i rapidi progressi nel settore grafica/video la NewTek deve continuare nella sua opera di aggiornamento, e la nuova versione offre infatti pionieristici sviluppi nelle tecniche di visualizzazione (alta risoluzione dinamica e HAM dinamico), oltre a significativi miglioramenti delle caratteristiche originali. Gli ingegneri della società di Topeka non avranno forse la bacchetta magica (non si può dire infatti che tutti gli aspetti della release 4.0 siano impeccabili), tuttavia i risultati complessivi rappresentano una vera pietra miliare nel cammino verso una videografica sempre più perfetta.

Alta risoluzione dinamica

Al primo sguardo, la caratteristica che colpisce di più di *DigiView Gold 4.0* è il modo "Dynamic HiRes", con la sua capacità di visualizzare contemporaneamente tutti i 4096 colori dell'Amiga su uno schermo in alta risoluzione e in overscan (768 x 480 pixel). Nell'alta risoluzione dinamica ciascuna linea orizzontale può avere una propria palette di 16 colori, quindi – almeno in teoria – in una sola schermata in alta risoluzione potrebbero essere presenti tutti i 4096 colori. Per variare i 16 colori di ogni linea, *DigiView* assume il completo controllo del 68000 e lo sincronizza con i coprocessori (il che significa, ovviamente, che durante la visualizzazione il multitasking viene sospeso). Persino muovere il mouse ha l'effetto di "sbriciolare" temporaneamente l'immagine. In tutti gli altri modi grafici, comunque, *DigiView* non intacca il multitasking dell'Amiga.

Per visualizzare le immagini nel modo Dynamic, la versione 4.0 contiene una speciale utility (*DynaShow*) che serve per visualizzare tutte le immagini IFF una per una oppure in sequenza. Probabilmente la NewTek è arrivata ai limiti estremi nello sfruttare le capacità di visualizzazione dell'Amiga standard, ma così facendo ha portato la macchina a una situazione di equilibrio precario... la vista è grandiosa ma il punto d'osservazione è piuttosto instabile.

Anche se un certo numero di programmatori stanno pianificando di aggiungere nei loro prodotti il supporto al modo Dynamic, a tutt'oggi non si può pensare di poter animare le immagini, colorarle, o comunque utilizzarle con la stessa flessibilità con cui si manipolano le immagini standard, Extra Half Brite o HAM. E, quel che più conta, non aspettatevi che basti accendere la videocamera per ottenere automaticamente i colori dell'HAM con la risoluzione 640 x 400 (parliamo della risoluzione americana, e non dei 640 x 512 pixel del PAL, perché pur funzionando anche in PAL fino ad oggi non ne è stata realizzata ancora una versione specifica, e quindi le immagini non riempiono tutto lo schermo).

La limitazione a 16 colori per linea di schermo spesso crea bande orizzontali sullo schermo, soprattutto dove esistono grandi variazioni di colore. Per esempio, se a sinistra ci sono il giallo, il rosso e il marrone, potrebbero non rimanere abbastanza colori a disposizione della linea per visualizzare a destra il blu e il verde.

Avendo lavorato con *DigiView* per quasi tre anni, mi considero ormai piuttosto esperto. Tuttavia, nessuna delle prime dieci immagini che ho digitalizzato in alta risoluzione dinamica si è rivelata del tutto immune dalle bande orizzontali. Ma evidentemente si tratta di un limite superabile via via che si acquisisce una perfetta padronanza del programma, come dimostrano le splendide immagini del disco "demo" di *DigiView 4.0*.

Il fatto che io abbia tentato per dieci volte senza successo di ottenere un'immagine "pulita" non significa che il verdetto nei confronti di questo sistema sia un inappellabile pollice verso. Significa soltanto che l'alta risoluzione dinamica è una sfida piuttosto ardua.

Il fattore critico è l'illuminazione,

che dev'essere accuratamente bilanciata. Per controllare la luminosità dei dati grafici primari, bisogna usare l'opzione Histogram del menu Project. Se è coperto almeno il 70 per cento della linea di base, allora probabilmente c'è abbastanza luce sul soggetto. Evitate i punti troppo luminosi perché moltiplicano le sfumature di colore percepibili dalla camera. Un altro fattore importante è la selezione delle immagini, perché ve ne sono molte di assolutamente irriproducibili in questo modo video. Viceversa, immagini ricche di colori e particolarmente dettagliate risultano facilmente riproducibili nel modo HAM 320 x 400. Inoltre, il normale dithering in alta risoluzione è superbo (il *dithering* è la mescolanza di punti di vario colore per ottenere l'impressione visiva di un colore diverso; i punti devono essere così piccoli e ravvicinati che l'occhio non li percepisce singolarmente, *ndr*). Secondo la mia personale opinione, sono poche le immagini per le quali l'alta risoluzione dinamica si dimostra una tecnica nettamente superiore alle altre.

In coppia con DigiPaint

Sarebbe inappropriato proporre una recensione di *DigiView* senza nemmeno citare il programma di disegno HAM della NewTek, *DigiPaint 3*. La possibilità di digitalizzare le immagini, infatti, ha un valore limitato se non esiste qualche mezzo per modificarle. Vi sono anche altri programmi di disegno che accettano le immagini prodotte da *DigiView*, ma *DigiPaint* ha in più il pregio di poter funzionare in contemporanea, rendendo possibile anche la modifica delle immagini in alta risoluzione dinamica. Una volta che un'immagine "Dynamic" è stata salvata, infatti, non è più possibile alterarla. *DigiView* mette quindi a disposizione il menu Controls, tramite il quale si può trasferire a *DigiPaint* un'immagine "catturata" e ritoccarla. Le informazioni RGB relative a un'immagine in alta risoluzione dinamica vengono elaborate in una superbitmap HAM; nel corso del lavoro di modifica in modo HAM, quindi, lo schermo non mostra l'immagine completa: per vederla bisogna ricorrere allo scroll. Il rituale che regola il complesso di queste operazioni è molto specifico, ma devo confessare che seguendo le indicazioni del manuale non ho avuto molto successo. Allora ho fatto qualche esperimento e, con l'aiuto dell'eccellente "supporto telefonico" della NewTek, sono arrivato a mettere insieme una procedura adeguata per la cattura, la modifica e il salvataggio di un'immagine dinamica.

Prima di tutto si deve escludere ogni task in background e mandare in esecuzione *DigiPaint*. Poi si deve mandare in esecuzione *DigiPaint*, impostare le dimensioni (Size) a 640 x 400 pixel, oppure a 768 x 480 in overscan. Si ritorna quindi al *Workbench* o al CLI e si manda in esecuzione *DigiView*, dopodiché si imposta l'appropriata dimensione di schermo HiRes. Si "cattura" quindi l'immagine e si eseguono tutte le regolazioni necessarie rimanendo nel modo a 16 colori. Quando si è soddisfatti di ciò che appare sul display, si preme il pulsante del mouse sull'opzione Dynamic del menu Controls. Ci vogliono circa dieci minuti perché l'immagine venga rielaborata in alta risoluzione dinamica. Nel frattempo lo schermo sembra pieno di coriandoli, finché all'improvviso non spunta fuori l'immagine finita. Soltanto a questo punto scopriamo se le nostre valutazioni, la scelta delle luci (e la fortuna) hanno trovato un giusto equilibrio.

A questo punto si seleziona *DigiPaint*, sempre dal menu Controls: compare lo schermo di quest'ultima applicazione, sul quale una versione HAM dell'immagine viene gradualmente ricostruita come superbitmap. Qui si può ritoccare l'immagine e salvarla. Si ritorna poi al modo Dynamic HiRes di *DigiView* e si carica l'immagine salvata. A questo punto si salva di nuovo l'immagine (questa

CHE COS'È DIGIVIEW

Il digitalizzatore DigiView *della NewTek è un pacchetto hardware/software che, da quando l'Amiga è nato, occupa i posti di vertice nella classifica delle applicazioni destinate alla videografica.*

Il primo DigiView *è stato progettato per l'Amiga 1000, e subito dopo è nato* DigiView Gold *per A500 e A2000. In breve i due programmi hanno acquistato un'immensa popolarità e sono diventati punti di riferimento obbligati nel mercato dell'Amiga.*

Anche se la versione 4.0, oggetto di questa recensione, è essenzialmente un upgrade della componente software *del digitalizzatore, pensiamo di fare cosa gradita ai neofiti descrivendo le caratteristiche generali del pacchetto.*

L'hardware di *DigiView* è una piccola unità che si collega alla porta parallela dell'Amiga. Il segnale video viene riversato nel computer tramite un jack RCA che si trova su questa unità. La "cattura" e l'elaborazione delle immagini vengono controllate grazie a un'eccellente interfaccia software.

DigiView porta a termine tre scansioni prima di costruire l'immagine finale, una per il rosso, una per il verde e una per il blu. Fonde poi i tre componenti, creando un'immagine IFF a colori.

Dell hardware fa parte anche un filtro-colore a forma di ruota, segmentato in quadranti di colore rosso, verde, blu e trasparenti. Il filtro s'installa davanti a una videocamera in bianco e nero, e permette di effettuare riprese a colori tramite un'opportuna rotazione in ogni successiva scansione. Un motorino (opzionale), battezzato DigiDroid ($79.95), fa ruotare automaticamente il filtro.

Altre società hanno prodotto congegni chiamati "color splitter", che sono in grado di separare i componenti rosso, verde e blu di un segnale video a colori. I risultati possono essere eccellenti, ma dipendono dalla qualità e dalla risoluzione del segnale a colori in ingresso, che in genere sono inferiori a quelle di una videocamera in bianco e nero.

Le selezioni di menu fanno comparire sullo schermo alcuni "pannelli di controllo" tramite i quali l'utente può selezionare la risoluzione, regolare l'allineamento (tracking), l'ampiezza e la posizione del segnale in ingresso, controllare la palette ed eseguire numerose variazioni sull'immagine prima di salvarla definitivamente. *DigiView* consente d'impiegare un'ampia varietà di risoluzioni (da 320 x 200 pixel fino all'alta risoluzione in overscan di 768 x 480 pixel). Si può anche variare il numero di colori utilizzati per l'immagine. I modi video permessi sono: bianco e nero, 32 colori, Extra Half Brite e HAM. La versione 4.0 introduce anche l'alta risoluzione dinamica e i modi HAM dinamici, di cui parliamo estesamente nell'articolo principale.

volta da *DigiView*) e si utilizza il Dyna-Show per visualizzarla come immagine in alta risoluzione dinamica. Senza questo secondo salvataggio, sarebbe stata visualizzata soltanto una parte della versione HAM.

C'è un'ultima ma importante osservazione da fare: si ricordi che sono necessari come minimo due megabyte di RAM per eseguire digitalizzazioni in tutti i modi possibili. Con una quantità di memoria inferiore non si possono eseguire digitalizzazioni in alta risoluzione 640 x 400 né in alta risoluzione dinamica. Per far funzionare *DigiView* e *DigiPaint* simultaneamente sono invece necessari almeno tre megabyte di RAM; perdipiù, non si possono modificare le immagini in alta risoluzione dinamica se i tre mega non includono il megabyte di chip RAM fornito dal nuovo Super Agnus "One-Mega". Suppongo che questa notizia non sia facile da digerire, ma dovete capire che continuando a lavorare con un mega costringete il vostro Amiga a rimanere molto al di sotto delle sue capacità potenziali, perché i programmatori – com'è ovvio – si sentono sfidati a progettare applicazioni che sfruttino sempre più a fondo soprattutto le configurazioni più potenti. In ogni caso, se anche non avete la possibilità di espandere il vostro Amiga, sappiate che anche al livello di un mega *DigiView* mette a disposizione eccellenti capacità di digitalizzazione ed elaborazione delle immagini. In genere anche un mega è più che sufficiente per soddisfare le esigenze dell'utente medio.

Un passo dopo l'altro

Anche se non prevedete di usare spesso il modo Dynamic, consiglierei ugualmente di acquistare l'upgrade alla nuova versione di *DigiView* (negli Stati Uniti è disponibile a 30.95 dollari). L'interfaccia è stata notevolmente migliorata, e vi sono anche altre modifiche significative. Il requester dei file, per esempio, elenca finalmente in ordine alfabetico tutti i dispositivi logici e i dischi disponibili. Scoprirete che è presente anche un miglior supporto al multitasking, compresa una barra di trascinamento che eventualmente si può disattivare e dotata di gadget per il riordinamento in profondità. C'è inoltre un'opzione open/close Workbench, che rende possibile un risparmio di memoria. La risoluzione, inoltre, può essere cambiata anche dall'interno di *DigiView* (ma in questo caso si perde l'immagine presente sullo schermo). La NewTek dichiara infine che il programma è perfettamente utilizzabile anche con il microprocessore 68020.

DIGIVIEW E AREXX, L'ACCOPPIATA VINCENTE

Ecco un esempio di file comandi *AREXX* destinato a controllare la ripresa automatica di fotogrammi a intervalli prestabiliti. Ci siamo limitati a due soli fotogrammi, ma crediamo che il listato dia un'idea abbastanza chiara di come opera l'*AREXX* e delle grandi possibilità che questo linguaggio fornisce agli utenti quando viene usato in coppia con applicazioni come *DigiView*

```
.key leavethishere

[illegible]
if warn
echo "Spiacente, questa demo richiede 1 megabyte
   di memoria"
wait 2
skip NoMemory
endif

if not exists RAM Echo
copy D Link RAM
copy C Wait RAM
copy C Echo RAM
endif

RAM echo "File comandi per la ripresa automatica
   di fotogrammi"
RAM echo "Per un minuto riprende un fotogramma
   in bianco e nero ogni 15 secondi"
RAM echo "Richiede 1 megabyte di memoria"
RAM echo ""

cd [illegible] e' necessario essere nella directory DigiView
RAM [illegible] controlla DigiView
[illegible]
RAM echo "[illegible] caricando DigiView"
[illegible]

nil: DV 320 200 o
RAM:wait 9          ;intervallo per il disco
                      cambiare il valore se e'
                      presente un hard disk

lab loop
RAM:DVLink mbaw
if warn
RAM:wait 2
skip
 loop back      ;attende che DigiView sia caricato
endif
endif

RAM echo "Digitalizzazione della prima immagine"
RAM:DVLink digr
RAM:DVLink scbk
;RAM:echo "Salvataggio di File1 in RAM"
;RAM:DVLink siff RAM:file1
RAM:echo "Sono in attesa di digitalizzare
   il fotogramma successivo"
RAM wait 5

RAM echo "Digitalizzazione della seconda immagine"
RAM.DVLink scfr
RAM.DVLink digr
RAM:DVLink scbk
;RAM:echo "Salvataggio di file2 in RAM"
;RAM DVLink siff RAM:file2
RAM:echo "Sono in attesa di digitalizzare
   il fotogramma successivo"
RAM wait 5

RAM:echo ""
RAM:echo "Ho finito "
RAM:DVLink quit

lab NoMemory
```

Una nuova caratteristica nell'elaborazione delle immagini, la "riduzione del rumore" (noise reduction), aiuta a contrastare la "grana", l'effetto neve e i disturbi ad alta frequenza del segnale video. Può anche controbilanciare l'assenza di grana causata dalla scarsità di luce. Quest'opzione è particolarmente utile quando viene usata in coppia con la regolazione della chiarezza dell'immagine (sharpness), e in questo caso i valori di riduzione del rumore più efficaci sembrano essere compresi fra 2 e 3. Se si va oltre, nell'immagine s'infiltra un aspetto artificiale e super-filtrato. Per quanto riguarda la "sharpness", raccomando di non superare il valore +5.

Una delle nuove aggiunte più significative, è sicuramente il supporto ad *AREXX*, un protocollo di comunicazione interprocessi che ha enormi implicazioni per il futuro dell'Amiga. L'*AREXX* si può definire un vero e proprio linguaggio, che mette l'utente in condizione d'impiegare un'applicazione per controllare alcuni aspetti di un'altra, o di scriversi da solo un programma di controllo. Tramite l'*AREXX*, quindi, si possono controllare tutte le funzioni di *DigiView* tramite un file comandi esterno. Per esempio c'è MHAM, che imposta il modo HAM; SARP(5), che imposta a +5 la sharpness, e così via. Vengono subito in mente applicazioni come la ripresa automatica di fotogrammi a intervalli prestabiliti, con conseguente elaborazione dell'immagine. Dal momento che anche *DigiPaint 3* è *AREXX*-compatibile, un'altra operazione che si potrebbe svolgere facilmente è il trasferimento automatico delle immagini al pacchetto di disegno, con conseguente elaborazione preprogrammata. Mi è capitato di parlare con un animatore che ha utilizzato questa tecnica per catturare, fondere e comprimere automaticamente i fotogrammi di un'animazione: tutto quello che doveva fare era cambiare la posizione degli oggetti in scena quando il programma lo avvertiva.

Questa integrazione di programmi è un passo avanti fondamentale, una vera e propria rivoluzione filosofica nella nostra consapevolezza su ciò che i computer rappresentano e su ciò che sono in grado di compiere, ed è entusiasmante vedere che moltissimi programmatori stanno aggiungendo alle loro applicazioni il supporto ad *AREXX*.

Il voto complessivo è senz'altro positivo, anche se la giuria mantiene una riserva sull'alta risoluzione dinamica. Tuttavia, la versione 4.0 aggiunge a *DigiView* significativi miglioramenti e vale ampiamente il prezzo richiesto per l'upgrade. *DigiView Gold 4.0* si colloca al primissimo posto tra i digitalizzatori Amiga, almeno per quanto riguarda le immagini di soggetti non in movimento. Per il resto possiamo osservare che l'interfaccia è ben progettata, il manuale è sintetico e completo, l'elaborazione delle immagini è eccellente e il prezzo è adeguato.

E bisogna anche ammettere che, se si segue religiosamente il rituale illustrato, anche l'alta risoluzione dinamica dà ottimi risultati... pur non essendo certo il Santo Graal dell'Hi-Res. ■

Per ulteriori informazioni contattare direttamente:

NewTek
(DigiView Gold 4.0, digitalizzatore + software $199.95, solo l'upgrade software $30.95)
115 W. Crane Street
Topeka, KS 66603, USA
(Tel. 001/913/3541146)

L'AMIEXPO 1990, DA NEW YORK A WASHINGTON

La Commodore tenta di dare il colpo di grazia all'Amiga 1000, e intanto si parla sempre più dell'AmigaDOS 2.0 che sarà installato direttamente nelle nuove ROM

di Morton A. Kevelson

A causa di circostanze non dipendenti dalla volontà degli organizzatori, quest'anno l'AmiEXPO anziché a New York si è tenuta a Washington, dal 16 al 18 marzo, all'hotel Sheraton. I tre giorni della mostra, a cui hanno partecipato 65 espositori, hanno visto la limitata partecipazione di 10.836 visitatori paganti. A mio parere, il notevole numero di mostre che ogni anno vengono dedicate all'Amiga ha prosciugato le tasche di parecchi potenziali partecipanti; come risultato, molti di loro hanno dovuto cominciare a scegliere con la massima attenzione a quali mostre recarsi. E il World of Amiga ha probabilmente dirottato l'attenzione di una parte del pubblico. Una cosa abbastanza curiosa è che, a quanto pare, è stata proprio l'assenza dell'AmiEXPO da New York che ha indotto l'Hunter Group di Toronto a organizzare in fretta e furia il World of Amiga a New York.

Comunque sono convinto che, indipendentemente dal momento e dal luogo in cui si svolge una mostra sull'Amiga, se un'azienda ha qualcosa di nuovo da presentare non rinuncia certo a partecipare. E infatti la maggior parte degli espositori presenti all'AmiEXPO aveva da proporre nuovi prodotti o miglioramenti di prodotti già esistenti.

Quando sono arrivato, mi sono reso conto ben presto che conveniva tenere gli occhi aperti non solo sugli espositori, ma anche sui partecipanti e sui vari stand di vendita. C'erano parecchi piccoli produttori che non avevano acquistato un loro personale spazio espositivo e vagabondavano per la mostra oppure s'installavano in un angolo dello stand di qualche venditore accomodante. Sarebbe stato facile non accorgersi nemmeno di loro, se non li avessi cercati tenendo le antenne all'erta.

Con la scheda Rejuvenator l'A1000 può accogliere l'ECS

Benché la Commodore stia attualmente facendo un ennesimo tentativo di eliminare l'Amiga 1000, c'erano almeno due produttori convinti che molti possessori di Amiga 1000 non avrebbero ingoiato l'esca della Casa di West Chester. Come nel caso della precedente offerta della Commodore (al tempo dell'uscita dell'Amiga 2000), oggi ai proprietari di un Amiga 1000 viene proposto l'acquisto di un Amiga 2000 per soli mille dollari, se riconsegnano il loro vecchio computer. Questa volta, però, per assicurarsi che dopo lo scambio i negozianti non rivendano gli Amiga 1000 agli ex-proprietari, la Commodore richiede che le tastiere vengano spedite a West Chester.

Non è molto chiaro quali siano i piani della Commodore per riutilizzare le vecchie tastiere, né quale sarà la sorte degli Amiga 1000 orfani della tastiera, ma come c'era da aspettarsi sta già comparendo un piccolo mercato per "adattatori di tastiere A2000 su console A1000". Forse la Commodore USA intende offrire le tastiere come accessori per il "super home computer per videogame" basato sull'Amiga, che a quanto si dice dovrebbe essere presentato ufficialmente in giugno al Consumer Electronics Show di Chicago.

In ogni caso, il tentativo di rastrellare tutti gli Amiga 1000 ancora in circolazione dipende dal fatto che queste macchine non saranno in grado di ospitare l'ECS (il nuovo set di chip dedicati), né le nuove ROM che conterranno la versione 2.0 dell'AmigaDOS. Dal momento che il nuovo Kickstart supera i 256K, non

sarà più compatibile con la memoria dell'Amiga 1000. Comunque, come dicevo, due progettisti hanno realizzato schede adattatrici che permetteranno d'installare l'ECS e il nuovo DOS negli Amiga 1000 superstiti.

All'AmiEXPO erano esposti alcuni esemplari della scheda di espansione "Rejuvenator" per l'Amiga 1000, prodotta dalla **Expert Services**. Il Rejuvenator sostituisce i 256K di WCS (memoria "Write Control Store", ovvero quella in cui si carica il Kickstart e che in seguito si comporta come una ROM) mettendola in grado di accettare il nuovo ECS e il nuovo Kickstart. Inoltre porta la RAM di sistema a 1,5 MB (1 megabyte di chip RAM). La scheda contiene inoltre un orologio che consente la visualizzazione del giorno e dell'ora e uno slot video simile a quello dell'A2000, ma non dispone di porta parallela o di output audio, per cui si deve ancora ricorrere alle porte situate sul retro della macchina. Il Rejuvenator costa 499,95 dollari, e comprende i chip di RAM ma non la ROM 2.0 e il nuovo Super Denise.

Un'altra possibilità di aggiornare il proprio A1000 viene dalla **Delaware Valley Software**. L'azienda non è riuscita a portare all'AmiEXPO un esemplare definitivo del suo DVS-Wonder, ma ha presentato un diagramma dell'unità. Il DVS-Wonder permette l'installazione del chip Super Agnus, mette a disposizione un orologio dotato di batteria tampone e può accettare fino a quattro ROM Kickstart consentendo di selezionarle via software. La scheda, inoltre, fa salire fino a 1 megabyte la quantità di chip RAM. Il DVS-Wonder sarà disponibile in diverse configurazioni, a partire dal ridottissimo DVS-Wonder-Kit (che comprende un circuito stampato e alcuni chip PAL, per $124,95), fino al completo DVS-Wonder-1024 (che include 1 MB di RAM, il nuovo Agnus, i Kickstart 1.2 e 1.3, un orologio in tempo reale e un filtro audio, $399,25). Dovrebbe essere messo in commercio anche il DVS-Wonder-Deluxe, che comprenderà tutte le caratteristiche elencate e in più 2 MB di chip RAM, il chip Super Agnus e il Kickstart 2.0, ma il prezzo non è ancora stato annunciato.

Presentandosi all'AmiEXPO, la **Computer System Associates** (CSA) ha sorpreso tutti. Questa società, che ha appena subito una totale riorganizzazione, era infatti scomparsa dal mercato all'improvviso e non aveva dato segni di vita per parecchio tempo. Il prodotto presentato dalla CSA si chiama Mega Midget Racer, ed è destinato ad A500 e A2000; l'unità si collega direttamente allo zoccolo del 68000, e mette a disposizione varie CPU 68030 che possono funzionare asincronicamente a 20, 25 o 33 MHz (e a ogni frequenza inferiore). I prezzi vanno dai 1399 dollari della versione da 20 MHz ai 1795 dollari di quella da 33 MHz.

Non mi è ancora capitato di partecipare a una mostra alla quale non fosse presente la **Great Valley Products**, e questa AmiEXPO non ha fatto eccezione. Il loro nuovo Impact SCSI + 8 (per l'A2000) offre un controller hard disk più 8 MB di RAM su un'unica scheda. Si può espandere la memoria a intervalli di 2 MB utilizzando SIMM da 1 MB. Il controller DMA include su scheda un suo personale buffer RAM da 16K. Il prezzo, RAM a parte, è di $399.

Nello stand della *Kick Start Magazine* mi sono imbattuto in Lawrence Hiller, della Chip Level Designs, che mi ha subito fatto vedere la nuova Nordic Power Cartridge (della **Data & Electronics**) importata dall'Olanda. Questa cartuccia, che si collega alla porta d'espansione dell'A500/A1000 o nello slot coprocessore dell'A2000, consente di congelare un programma in esecuzione e di salvare su disco fino a 1 MB di memoria. La cartuccia contiene anche alcune routine in ROM che forniscono un monitor per il linguaggio macchina, la capacità di cercare in memoria immagini e suoni campionati (salvandoli poi come file IFF) e un set completo di utility per i dischi. La cartuccia Nordic Power è dotata di una ROM da 64K e di una RAM da 128K. Prezzo e data dell'uscita non sono ancora stati fissati.

Un vociante gruppo di "fornai" della **Blue Ribbon Bakery** illustrava l'ultima versione di *Bars&Pipes*, il primo programma musicale e MIDI per l'Amiga basato su icone e "object oriented".

Ma ci sono anche altre novità. Per esempio cominciano a fare la loro apparizione mezzi di lettura/scrittura ottici anche per l'Amiga. La **Computerall Services** esponeva un disk drive ottico da 650 MB: Tidal Wave. Si tratta di un dispositivo a lettura/scrittura, rimovibile, che viene venduto a un prezzo di 4995 dollari. La Computerall esponeva inoltre il nuovo programma *Autoscript* ($129,95) che "traduce" file PostScript in una forma comprensibile ad applicazioni come *Sculpt 4D* o *Turbo Silver*, consentendo quindi di utilizzare con questi pacchetti l'output prodotto da programmi grafici come *Adobe Illustrator 88*, *Aldus Freehand*, *LetraStudio* e *Professional Draw*.

Anche la **Impulse** ha inserito tra i suoi prodotti un disk drive ottico (600 MB, lettura/scrittura), battezzato MegaDrive. A sua volta, la **Intuitive Technologies** esponeva la versione 1.9 di *Ultracard Plus*, un software che veniva fatto funzionare in coppia con un drive per CD ROM ad alta capacità.

L'AmiEXPO non sarebbe più se

Sopra: la scheda Mega Midget Racer per A500/A2000. Sotto: conferenza di Jay Miner, il "padre" dell'Amiga

stessa senza la tradizionale serie di prestigiosi conferenzieri, e anche quest'anno la consuetudine è stata rispettata. Sono riuscito ad assistere a un intervento di Jay Miner; la relazione, che era intitolata "L'Amiga: dal principio", si è rivelata in realtà una serie di considerazioni di Miner su ciò che – secondo lui – tutti dovrebbero sapere sull'Amiga. Essendo stato il responsabile della realizzazione dei chip dedicati su cui è basato l'Amiga, Jay Miner è considerato il "padre" di questo computer.

Nel suo discorso, Miner ha suggerito parecchie strade da seguire per pubblicizzare maggiormente l'Amiga, compresa l'idea di lasciare copie di vecchie riviste sull'Amiga nelle sale d'attesa dei medici. Una voce dal pubblico ha suggerito di lasciarne qualche copia anche nei gabinetti dei dirigenti delle società più importanti: un buon modo per garantirsi dieci minuti d'ininterrotta lettura da parte degli alti livelli manageriali.

Come accade per tutte le mostre di computer, l'AmiEXPO si è conclusa con una serie di acquisti e di vendite sempre più frenetica, a mano a mano che gli sconti proposti dai venditori diventavano più consistenti (nel tentativo di riportare indietro la minor quantità possibile di materiale). L'esempio più notevole di questa sindrome è stato offerto dagli stessi promotori dell'AmiEXPO, che si sono messi a "vendere" *Light Year* (calendario del 1990 con immagini realizzate tramite l'Amiga) con uno sconto del cento per cento sul prezzo di copertina, che era di 14,95 dollari. Un affare più che discreto per un calendario dell'anno in corso, considerando che eravamo in marzo. ■

Per ulteriori informazioni:

Expert Services
5912 Centennial Circle
Florence, KY 41042, USA
(tel. 001/606/3719690)

Delaware Valley Software
P.O. Box 2007
Upper Darby, PA 19082-0507, USA
(Tel. 001/215/4469227)

Data & Electronics
P.O. Box 3153
NL-5902 RD Venlo, Olanda

Computer System Associates
7564 Trade Street
San Diego, CA 92121, USA
(Tel. 001/619/5666911)

Great Valley Products
225 Plank Avenue
Paoli, PA 19301, USA
(Tel. 001/215/8899411)

Blue Ribbon Bakery
1248 Clairmont Road, Suite 3D
Atlanta, GA 30030, USA
(Tel. 001/404/3771514)

Computerall Services
3 North Walnut Avenue
New Hampton, Iowa 50659, USA
(Tel. 001/515/3943778)

Impulse Inc.
6870 Shingle Creek Parkway 112
Minneapolis, Minnesota 55340, USA
(Tel. 001/612/5660221)

Intuitive Technologies
1199 Forest Avenue, Suite 264
Pacific Grove, California, USA
(Tel. 001/408/6469147)

I PROTOCOLLI DI TRASFERIMENTO DALLA X ALLA Z

Ecco la prima parte di una serie di articoli dedicati ad algoritmi e tecniche di collegamento nei protocolli X-Y-ZModem, con particolare riferimento alla correzione degli errori

di Avelino De Sabbata

Nell'articolo pubblicato sul numero 4/89 di *Commodore Gazette* ci siamo già soffermati sulla difficoltà di rilevare e correggere gli inevitabili errori che si verificano nel trasferimento dei dati. In quell'occasione è stata analizzata la semplice procedura XON-XOFF e sono stati esposti in modo schematico gli algoritmi dei protocolli di trasferimento ASCII, XON-XOFF, Kermit e XModem.

Con questo articolo e con quelli che lo seguiranno torniamo a parlare di protocolli di trasferimento dati. Partiremo dal semplice ma ancora attuale algoritmo XModem proposto nel 1977 da Ward Christensen (che avremo modo di esaminare in profondità, analizzandone anche gli aggiornamenti nelle svariate applicazioni), ci soffermeremo poi sull'YModem, diretto discendente del protocollo di Christensen, e arriveremo infine al sofisticato ZModem, il quale riunisce in sé le prerogative migliori dei suoi predecessori, colmandone nello stesso tempo molte lacune.

Gran parte delle nostre fonti sono costituite dalla documentazione ufficiale redatta nel 1977 da Ward Christensen per quanto riguarda l'XModem, mentre per quanto concerne le estensioni al protocollo originale e gli algoritmi utilizzati nello ZModem, le notizie sono state tratte in gran parte dal libro *The ZModem Inter Application File Transfer Protocol* e da un compendio alla documentazione originale, *XModem/YModem Protocol Reference*, entrambi redatti nel 1987 da Chuck Forsberg della Omen Technology Inc.

Iniziamo illustrando la logica di base del protocollo XModem (come la intende Christensen), soffermandoci in particolare sull'uso che gran parte delle applicazioni fanno del carattere CAN, non previsto dal protocollo originale, ma di fatto inserito nello standard XModem. Ci scusiamo in anticipo per le ripetizioni di concetti già esposti nel primo articolo della serie Computer e Trasmissione Dati, ma si tratta dell'unico modo per evitare fastidiosi rimandi.

I protocolli X, Y e ZModem prevedono, per la sincronizzazione dei terminali e per lo scambio di segnali, l'uso di alcuni caratteri di controllo la cui funzione diventerà del tutto chiara procedendo nella trattazione.

I caratteri di controllo, nel corso dell'articolo, saranno indicati nella classica forma che prevede il prefisso "^" al carattere da premere assieme al tasto CTRL. Se per esempio s'incontra il simbolo "^A" s'intende indicare il codice ASCII ottenibile premendo il tasto "A" contemporaneamente al tasto Control, di solito contrassegnato con "CTRL" sulla tastiera.

XModem

Il protocollo prevede un collegamento asincrono, a 8 bit, senza parità e con un bit di stop. Dal momento che non viene imposta alcuna restrizione sul contenuto dei dati trasmessi, possono essere trasmessi dati in qualsiasi formato: binario, ASCII e così via.

In genere i file da trasferire non subiscono modifiche, eccetto un'eventuale aggiunta in coda di alcuni caratteri NULL (il cosiddetto "padding"), che di solito non influisce sul funzionamento dei programmi eseguibili o sui file di testo. Il padding ha invece serie conseguenze per i file nei quali la lunghezza in byte è determinante.

Le versioni di XModem create per i sistemi operativi CP/M e MS-DOS hanno una particolarità: necessitano di un carattere CTRL-Z (ASCII $1A) come segnalatore di fine file (EOF), che non serve invece negli altri sistemi. È comunque consigliabile estendere l'uso del terminatore ^Z a tutte le applicazioni, in quanto ciò rende possibile l'eliminazione del padding (tramite l'impiego di utility esterne o anche tramite lo stesso programma di trasmissione). Anche per l'Amiga esistono varie utility PD che si occupano di "ripulire" dai caratteri estranei i file ricevuti con XModem (come per esempio FIXOBJ).

Riportiamo le caratteristiche che un file di testo ASCII dovrebbe avere per potersi definire standard, e

mantenere quindi la compatibilità con tutti i sistemi.

- Carattere per la tabulazione: ASCII $09; un TAB impostato ogni otto caratteri.
- Terminatore di linea (EOL) formato dai caratteri CR/LF ($0D $0A).
- Fine del file (EOF) indicata da uno o più caratteri ^Z.
- File di lunghezza variabile divisi in blocchi di 128 byte.
- Unicamente per il CP/M esiste ancora una peculiarità: se i dati formano un multiplo esatto di 128 (e quindi il file trasmesso secondo il protocollo XModem contiene l'ultimo terminatore di linea LF nel 128° byte del blocco e il CR nel 127°), il blocco seguente che conterrebbe soltanto caratteri ^Z è opzionale (ma è consigliabile inserirlo ugualmente, dal momento che alcune utility non riconoscono la fine del file in assenza del/dei caratteri ^Z).

Organizzazione dei dati

I dati da trasmettere vengono sempre suddivisi in blocchi di 128 byte, secondo lo schema seguente:

```
[SOH] [BLK#] [255-BLK#] [128 BYTE DI DATI] [CKSUM]
```

SOH è il carattere che individua l'inizio di ogni blocco.

BLK# è un contatore sequenziale (lungo un byte) che contrassegna il numero di blocco e che va da 1 a 255. Raggiunto il valore 255 salta a 0, (non a 1), come nel seguente esempio: 1-2-3... 254-255-0-1-2...

255-BLK# è il complemento a 1 del numero di blocco (XOR #BLK).

DATI è un blocco di dati lungo 128 byte. Se l'ultimo blocco del file contiene meno di 128 byte, vengono aggiunti tanti caratteri NULL quanti ne servono per raggiungere il valore 128. In genere la fine del file è segnalata da un carattere ^Z.

CKSUM è un byte di controllo che viene ottenuto sommando tutti i byte dei dati senza fare uso del riporto: CKSUM=(CKSUM+DATA) AND 255.

Il protocollo prevede una prima fase di sincronizzazione dei terminali, durante la quale il trasmittente attende per un tempo variabile da dieci a sessanta secondi l'arrivo di un carattere NAK, il quale segnala che il ricevente è pronto a ricevere i dati (è consigliabile mantenere al massimo il limite di timeout per la sincronizzazione, lasciando che sia il ricevente a guidare il trasferimento: RECEIVER-DRIVEN). In questa fase, il trasmittente ignora qualsiasi carattere diverso da NAK o CAN. La ricezione del carattere CAN provoca l'interruzione del trasferimento.

Il ricevente invia un NAK e attende per dieci secondi l'arrivo di un carattere SOH; la procedura viene eventualmente ripetuta fino a un massimo di dieci volte, dopodiché il trasferimento non può più avvenire. Sullo schermo dovrebbe comparire il seguente messaggio di errore: "ABORTED: NAK retry maximun exceeded!".

Portata a termine la fase di sincronizzazione, il trasmittente invia il primo pacchetto e si dispone in attesa. Il destinatario calcola il checksum del pacchetto e lo confronta con quello ricevuto: se corrispondono invia un ACK, in caso contrario segnala la condizione di errore con un NAK.

Nel primo caso il trasmittente provvede a inviare il pacchetto successivo, altrimenti ritrasmette lo stesso pacchetto.

Durante la trasmissione dei dati e del byte di checksum, il timeout (massimo tempo d'attesa di un carattere) viene impostato dal ricevente a 1 secondo, e torna a essere impostato a dieci secondi solo durante l'attesa della testata del blocco successivo.

Dopo aver inviato l'ultimo pacchetto e aver ricevuto il corrispondente ACK, il trasmittente manda un carattere EOT per segnalare la fine della trasmissione e si pone in attesa dell'ultimo ACK. A questo punto il ricevente, individuando un carattere EOT al posto del SOH che segnalerebbe l'inizio di un nuovo pacchetto, chiude il file e ne dà conferma con un ultimo ACK.

Nel diagramma che segue è rappresentato il flusso di un trasferimento senza errori, guidato dal protocollo XModem.

TRASMITTENTE	***RICEVENTE***
	<-- [NAK]
[SOH] [$01] [$FE] [128 DATA] [CKSUM] -->	
	<-- [ACK]
[SOH] [$02] [$FD] [128 DATA] [CKSUM] -->	
	<-- [ACK]
[SOH] [$03] [$FC] [128 DATA] [CKSUM] -->	
	<-- [ACK]
[EOT]-->	
	<-- [ACK]

Errori e relative correzioni

Il protocollo XModem ha principalmente lo scopo di trovare e correggere gli errori di trasmissione. Il ricevente e il trasmittente tentano autonomamente di recuperare gli errori facendo fino a dieci tentativi consecutivi per ogni pacchetto. Il documento originale consiglia d'interpellare l'utente dopo dieci tentativi, chiedendogli se si deve abbandonare il trasferimento in corso oppure continuare nei tentativi (vi sono alcune applicazioni che seguono questa indicazione). In realtà, quasi tutti gli errori recuperabili vengono eliminati nelle prime prove. Se la linea è così disturbata che dieci tentativi non sono sufficienti, conviene troncare il collegamento in corso e rifare la chiamata.

Falsa condizione di CAN. Il carattere CAN (^X) serve per interrompere il trasferimento dati in corso. Se il trasmittente riceve questo carattere mentre è in attesa di un ACK (o di un NAK), annulla il trasferimento e abbandona il protocollo XModem. Allo stesso modo si comporta il ricevente se accerta l'arrivo di un CAN mentre aspetta un SOH (oppure un EOT).

Però a volte il carattere CAN viene generato da disturbi nel collegamento, ed è quindi consigliabile determinare l'interruzione del trasferimento tramite il

ricevimento di due CAN consecutivi (ed è ciò che infatti fanno molte versioni di XModem).

Falsa condizione di EOT. Anche per il segnalatore di fine trasmissione valgono le considerazioni esposte nel caso del CAN. Il carattere SOH inviato in testa al pacchetto potrebbe infatti essere erroneamente interpretato come carattere EOT, costringendo il ricevente a chiudere la trasmissione (di solito accade quando ricevente e trasmittente hanno una sincronizzazione leggermente diversa). È dunque consigliabile dotare i programmi di un algoritmo capace di prevenire false condizioni di EOT; come nel caso precedente, il sistema più semplice è quello di assumere come condizione di fine trasmissione il ricevimento di due EOT consecutivi. Al primo EOT il ricevente risponde con un NAK, ma se ne riceve un altro sa con certezza che è giunto il segnale di fine file. Nel caso che il trasmittente non fosse predisposto per inviare il secondo EOT, è consigliabile impostare il timeout a tre secondi e ritrasmettere il NAK fino a tre volte di seguito. Non ricevendo alcuna risposta si accetta la chiusura del trasferimento, inviando però un opportuno messaggio di avvertimento.

Errori di tempo limite in ricezione e in trasmissione. Questi errori sono già stati trattati, ma ne riassumiamo comunque le condizioni.

Se durante l'attesa dei dati trascorrono dieci secondi senza che arrivi nessun carattere, il ricevente invia un NAK. Questo si può ripetere fino a dieci volte consecutive, dopodiché il trasferimento viene annullato. Durante il trasferimento dei dati il timeout è impostato a un secondo.

Il trasmittente, secondo il protocollo originale, una volta inviato un pacchetto attende per dieci secondi l'arrivo di un carattere ACK, NAK o CAN, dopodiché ritrasmette l'ultimo pacchetto come se avesse ricevuto un NAK. La procedura viene ripetuta per dieci volte, e alla fine il trasferimento viene interrotto. In molte applicazioni i dieci secondi di attesa del trasmittente consigliati dal protocollo originale sono stati aumentati, arrivando fino a 30/60 secondi.

Errori nei byte di numerazione del pacchetto. Se il numero di sequenza è uguale a quello dell'ultimo pacchetto ricevuto senza errori, significa che il carattere ACK inviato dal ricevente non è stato interpretato correttamente, e pertanto il pacchetto in arrivo dev'essere ignorato e si deve inviare al trasmittente un semplice carattere ACK.

Errore di sequenza. Quando il numero di sequenza (BLK#), combacia con il complemento a 1 del byte successivo (255-BLK#), ma non corrisponde al numero di sequenza precedente aumentato di 1, il protocollo originale dà per scontato che sia avvenuta una perdita irreparabile di uno o più pacchetti, e pertanto consiglia l'invio di due o più caratteri CAN per interrompere il trasferimento. Esiste però anche la possibilità che due successivi byte BLK# assumano valori errati, ma casualmente complementari. Pertanto consiglieremmo di non escludere a priori questa possibilità e d'intraprendere un tentativo di recupero inviando il solito NAK.

Errore nel numero complementare. Quando il numero di sequenza (BLK#) non è uguale al complemento a 1 del byte successivo, è avvenuto un errore nel trasferimento di uno dei due byte e pertanto il pacchetto viene ignorato; al trasmittente viene inviato un carattere NAK.

Errori di sincronizzazione. Durante le comunicazioni asincrone vengono spesso generati caratteri estranei al trasferimento, ed è quindi necessario che il ricevente abbia la possibilità di rilevarlo. Questo si può ottenere tenendo il conto dei caratteri ricevuti, contando per ogni pacchetto un totale di 132 caratteri. Se i terminali non risultano più sincronizzati, significa che nel pacchetto si sono introdotti caratteri estranei, e conviene quindi procedere a un'ulteriore fase di sincronizzazione, analoga a quella iniziale.

La ricerca dell'inizio del pacchetto si effettua accertando l'arrivo di un SOH, e ignorando qualsiasi altro carattere. Al sopraggiungere di un SOH il ricevente ipotizza che i due caratteri successivi siano il numero di sequenza e il relativo carattere di controllo. Se questi due valori sono complementari e corrispondono al numero di sequenza atteso, si dà per scontato che quello sia l'inizio del pacchetto atteso.

Se durante la fase di ri-sincronizzazione viene superato il tempo limite (che il ricevente avrà cura d'impostare a dieci secondi), è necessario inviare un NAK. Il protocollo prevede inoltre che in questa fase si invii un NAK anche dopo il ricevimento di 135 caratteri, se non è stata individuata la testata di un pacchetto. È consigliabile che quest'ultimo accorgimento sia adottato anche durante la sincronizzazione iniziale.

Un esempio

Nel diagramma che segue è rappresentato il flusso di un trasferimento che contiene alcuni errori. Si osservi come XModem ne effettua il recupero.

TRASMITTENTE	***RICEVENTE***
	<— [NAK]
(Non invia alcun carattere)	(10 secondi)
	<— [NAK]
[SOH] [$01] [$FE] [128 DATA] [CKSUM] —>	
	<— [ACK]
[SOH] [$02] [$FD] [128 DATA] [CKSUM] —	
(viene rilevato un errore)	
	<— [NAK]
[SOH] [$02] [$FD] [128 DATA] [CKSUM] —>	
	<— [ACK]
[SOH] [$03] [$FC] [128 DATA] [CKSUM] —>	
(ACK non viene riconosciuto)	
	<— [ACK]
[SOH] [$03] [$FC] [128 DATA] [CKSUM] —>	
(Il pacchetto viene ignorato)	
	<— [ACK]
[EOT]—>	
	<— [NAK]
[EOT]—>	
	<— [ACK]

Nel prossimo numero l'analisi dei protocolli di comunicazione e dei relativi errori continuerà con le migliorie apportate a XModem e in seguito con YModem e ZModem.

■

MUSICA E COMPUTER

LA COMPUTERMUSICA BUSSA ALLE PORTE DEL DUEMILA

di Massimiliano Lisa

Un breve sguardo sugli anni Novanta, che porteranno alle Personal Workstation e a una musica sempre più intimamente legata ai computer. Accanto a questa proiezione verso il futuro, continua la nostra rubrica dedicata alle novità per l'Amiga

Siamo ormai entrati negli anni Novanta, e ci troviamo sulla soglia di nuovi mutamenti che renderanno la musica un'arte ancor più intimamente legata alla tecnologia.

Già oggi è disponibile una nuova generazione di computer, quella delle macchine a 32 bit basate sui microprocessori Motorola 68030/68040 e Intel 80486. Sono macchine come i Mac SE/30, IIx e IIcx della Apple, l'Atari TT, i PC IBM e compatibili, il NeXT e l'Amiga 3000.

Il mondo dei 16 bit consentiva già l'elaborazione dei dati MIDI, ma i 32 bit aprono la porta a qualcosa che sino a oggi non era ancora realmente compiuto: l'audio digitale e il multitasking.

Le macchine a 32 bit sono in grado di operare a una velocità almeno doppia rispetto alla generazione precedente, dal momento che possono leggere ed elaborare blocchi di dati due volte più grandi. La velocità di clock più elevata e la capacità d'indirizzare direttamente maggiori quantità di memoria si traducono quindi in una superiore capacità d'elaborazione, e questo è fondamentale nelle applicazioni musicali. Sequencer di alto livello, programmi di notazione e di campionamento sono applicazioni legate intimamente alle caratteristiche del microprocessore, e un aumento di velocità vuol dire molto.

In secondo luogo, la velocità di elaborazione raggiungibile con le macchine a 32 bit inizia ad avvicinarsi a quella necessaria per un campionamento audio di buon livello. Possiamo dire che ci stiamo lasciando indietro l'era dei PC (Personal Computer) per entrare in quella delle PW (Personal Workstation).

Un'altra importante differenza rispetto al passato è che i nuovi microprocessori dispongono di piccole quantità di RAM "private", chiamate *cache buffer*, nelle quali possono caricare porzioni di codice del programma e dati. I cache buffer di istruzioni permettono al microprocessore un caricamento diretto delle istruzioni, evitandogli di passare ogni volta dalla memoria centrale.

Un cache buffer di dati, invece, contiene i dati che si usano con maggior frequenza, il che ancora una volta contribuisce a migliorare le prestazioni.

All'aumentare della potenza dei computer, però, corrisponde un aumento della necessità di memoria. Se far funzionare il *Presentation Manager* dell'OS/2 richiede 4 MB di RAM e un hard disk, anche elaboratori come l'Amiga non sono da meno. L'Amiga 3000 ha infatti una RAM di 2 MB e l'hard disk di serie.

Ma per il mondo musicale non basta ancora. Trentadue megabyte vi sembrano tanti? Be', considerate questo: con 32 MB si possono registrare digitalmente – a 44,1 KHz – la bellezza di tre minuti in stereo, un minuto e mezzo a quattro tracce e 45 secondi a otto tracce. E ricorrere alla memoria virtuale non è particolarmente facile (parliamo di quella tecnica che fa credere al computer di avere più RAM di quella che in realtà possiede, grazie a un rapido passaggio di dati tra RAM e hard disk). Su molte macchine infatti questa tecnica è utilizzabile con efficacia soltanto per applicazioni come quelle di disegno, ossia per applicazioni non troppo sensibili alle temporizzazioni. Con la musica, invece, il tempo è cruciale.

Registrare un audio stereo con la qualità di un CD richiede circa dieci megabyte per ogni minuto di musica. È evidente quindi che i dischi rimovibili da 150-400 MB sono un punto d'arrivo obbligatorio.

Nei prossimi dieci anni possiamo aspettarci di vedere computer sempre più potenti, vere e proprie Personal Workstation, che entreranno nella vita di tutti i musicisti. A capacità sempre maggiori corrisponderà la possibilità di manipolare i suoni a nostro piacimento, di sostituire completamente la registrazione analogica su nastro magnetico con quella digitale su hard disk e di arrivare a capacità di edit che non porranno alcun limite alla fantasia.

Per iniziare a sintonizzarci col futuro, vi proponiamo una serie di opinioni di esperti del settore (*brani tratti da interviste rilasciate alla rivista* Electronic Musician).

«Nei concerti dal vivo faranno la loro apparizione musicisti elettronici che in parte saranno programmati dagli artisti, e in parte, attraverso l'Intelligenza Artificiale, disporranno di una conoscenza musicale sufficiente per improvvisare insieme ai musicisti in carne e ossa. [...] Le composizioni del ventunesimo secolo non saranno più dischi da ascoltare e basta. L'artista fornirà una serie di strutture espressive che consentiranno all'ascoltatore d'interagire con la musica e di far eseguire il pezzo ogni volta in modo diverso».
Raymond Kurzweil

«Si arriverà a CD MIDI che permetteranno di collegare i propri strumenti via MIDI perché suonino insieme al materiale registrato su disco». **Dave Kusek**

«Saranno disponibili mixer e registratori digitali coi quali si potrà avere un controllo totale sulla musica. Si potranno manipolare i suoni a piacimento ed effettuare montaggi audio oggi impensabili». **Larry Fast**

«Il prossimo passo consisterà nel sostituire i sintetizzatori su rack con schede da inserire nei computer. [...] Il MIDI 1.0 è superato. È tempo che si passi al MIDI 2.0». **Dave Oppenheim**

«Si passerà da un mondo musicale analogico a un mondo totalmente digitale. Uno studio casalingo multitraccia verrà a costare meno di diecimila dollari. E questo avverrà prima della fine degli anni Novanta». **Marcus Ryle**

NOVITÀ PER L'AMIGA

Una rapida panoramica su Tiger Cub, Bars&Pipes e Master Tracks Pro

Tiger Cub della Dr. T's raccoglie in un unico pacchetto, sequencer, editor grafico e sistema di notazione, a un prezzo inferiore ai cento dollari (richiede almeno 1 MB). Oltre ai suoni ottenibili collegando strumenti via MIDI, si possono utilizzare anche le voci interne del computer.

Il programma è diviso in due schermi principali: sequencer ed editor grafico di eventi. Il sequencer è in grado di gestire fino a 12 tracce con una risoluzione di 384 impulsi per nota da un quarto. Tutte le regolazioni possono essere modificate in tempo reale (alterazione dei tempi, ricanalizzazione, possibilità di filtrare gli eventi...).

L'editor grafico visualizza una traccia alla volta, con il tempo raffigurato sull'asse X e l'altezza della nota sull'asse Y. Le note sono rappresentate come simboli a forma di L. La posizione della linea di base e la lunghezza rappresentano il tempo d'inizio, l'altezza e la durata, mentre la lunghezza della gamba stabilisce la velocità di attivazione della nota. Una finestra mostra altri tipi di eventi MIDI. Sempre dall'editor grafico è possibile inserire nuove note e modificare un evento o gruppi di eventi. L'edit è esteso a opzioni come copia, taglia, unisci, cancella e disfa. Non manca la possibilità di modificare altezza, velocità, quantizzazione e tempo d'inizio.

Il pacchetto è completato da un'utilissima funzione che consente di visualizzare e di stampare la musica con la notazione standard. È possibile vedere sul monitor e stampare su carta una singola traccia o l'intera partitura. Alcune particolari opzioni consentono di modificare lo stile delle gambe delle note, la quantizzazione e la trasposizione. Tra le stampanti supportate troviamo le Epson FX/LX, i modelli a 24 aghi della NEC e della Panasonic, e le HP DeskJet e InkJet. È possibile leggere e scrivere file in formato MIDI e KCS.ALL.

Nella fascia di prezzo alla quale il prodotto appartiene (costa solo $99) non esiste in pratica concorrenza. Tiger Cub offre le caratteristiche di un sequencer decisamente più costoso, e in più mette a disposizione un editor grafico e un sistema di notazione. È il prodotto ideale per chi inizia a lavorare con la computermusica e non vuole spendere troppo.

Bars&Pipes. Si tratta di un sequencer per l'Amiga che offre tutte le caratteristiche standard e in più alcune opzioni particolari. Ciò che lo distingue, però, sono la sua interfaccia grafica e la sua concezione progettuale. Il tutto è infatti basato su un'originale metafora "idraulica": ogni traccia costituisce un condotto in cui scorrono i dati MIDI. L'utente è come un idraulico e gestisce a suo piacimento i flussi di dati.

Il cuore del programma è la PipeLine (oleodotto). I dati MIDI in ingresso entrano nello schermo da sinistra, possono essere convogliati verso qualsiasi traccia, registrati e rivisualizzati sullo schermo. Alla fine di ogni condotto (pipe) si può decidere tramite un rubinetto se far passare o se bloccare i messaggi MIDI (ovvero eventi, velocità, pitch bend, aftertouch, program change...).

I dati possono essere registrati, riascoltati in riproduzione, ridirezionati e fusi. Elaborazioni come la quantizzazione, l'impostazione di filtri e la trasposizione possono essere effettuate sia in fase di input che in fase di output.

Nell'impostazione di default non importa su quale canale lo strumento invia i dati, dal momento che il programma ascolta tutti e 16 i canali. Per chi si diletta di applicazioni MIDI legate a controller per chitarra, la caratteristica di multicanale e il modo mono consentono di assegnare ogni corda a una traccia separata.

La quantità di memoria richiesta è di almeno 1 MB. *Bars&Pipes* non pone limiti al numero di tracce e alla loro lunghezza: l'unico limite è la memoria.

Gli strumenti a disposizione dell'idraulico/musicista sono una trentina. Si va da MIDI echo e delay, a keyboard split, quantize, unquantize, articulator (durata delle note), transpose, modulate... Non mancano utility come ***Counterpoint*** e ***Harmony***, che sulla base di algoritmi pre-programmati e della tonalità designata, generano nuovo materiale partendo dalle note ricevute. ***Triad*** invece, produce accordi aggiungendo due note a tutte le note in ingresso. ***Sforzando*** conferisce un'insolita dinamicità alle note inviando eventi polifonici di aftertouch. ***Phrase Sharper*** individua le variazioni di dinamica specificate nella finestra Master Parameters e modifica di conseguenza la velocità delle note. ***Reverser*** suona una sequenza all'inverso. ***Note Flipper*** crea una melodia basandosi su una segnatura. Ma il tool che brilla su tutti gli altri è ***Accompany B***, che esegue improvvisazioni sulla base dei ritmi e degli accordi specificati nella finestra Song Parameters.

I vari tool possono essere attivati e regolati in tempo reale. È per esempio possibile attivare la quantizzazione del charleston di una batteria e regolarla mentre si ascolta la canzone, finché non si ottiene l'effetto desiderato. Se l'utente trova una configurazione di lavoro particolarmente adatta alle sue esigenze (basata su specifici tool, regolazioni e così via), può salvarla facendola diventare una macro e disegnando un'apposita icona per poterla poi richiamare.

La finestra per l'edit delle tracce offre due tipi di rappresentazione: una griglia stile tastiera di pianoforte e una sorta di rigo musicale sul quale le note e la loro durata sono rappresentate da barre. È anche possibile visualizzare pitch bend, velocità, aftertouch, controller e program change. Le funzioni di edit sono semplici e rapide: c'è una penna per gli inserimenti e una gomma per le cancellature. Le varie operazioni si possono compiere sia sulla tastiera sia sul rigo musicale.

Si tratta di un sistema di rappresentazione abbastanza efficace e certamente utile a chi conosce poco di teoria musicale. Una rappresentazione delle note con la consueta notazione, comunque, avrebbe avuto anche i suoi vantaggi: non è poi così immediato distinguere tra loro note da 1/16 e da 1/8 osservando solo la lunghezza della riga.

La risoluzione del clock del programma è di 192 impulsi per nota da un quarto, ossia lo standard di molti altri sequencer per l'Amiga. Va però rilevato che è decisamente più bassa di quella fornita da sequencer realizzati per altri computer, e anche di quella di *Tiger Cub*. All'esame, il clock si è dimostrato abbastanza affidabile durante movimenti del mouse e apertura di finestre, ma molto meno durante il multitasking o le operazioni su disco. Il clock può essere visualizzato anche in tempo SMPTE (ore, minuti, secondi e frame).

In definitiva, ci troviamo dinanzi a un programma complesso, ma accessibile tanto al neofita quanto al professionista. L'aspetto che colpisce di più è l'approccio di tipo "idraulico" alla musica: potreste trovarlo eccezionale, o assolutamente indegno di paragone con un modo di operare alla *KCS* della Dr. T's. Non vi resta che azionare i rubinetti dell'oleodotto, per scoprirlo...

Master Tracks Pro. Concludiamo il nostro appuntamento con la computermusica segnalando la disponibilità di *Master Tracks Pro* anche per l'Amiga. Il sequencer offre 64 tracce, il Song Editor Passport e un editor grafico. È possibile utilizzare il codice SMPTE. ■

Lo schermo "idraulico" di Bars&Pipes: a destra ci sono i rubinetti

Per ulteriori informazioni contattare direttamente:

Blue Ribbon Bakery
(Bars&Pipes, $299)
1248 Clairmont Rd.
Suite 3D
Atlanta, GA 30030, USA
(Tel. 001/404/3771514)

Dr. T's Music Software
(Tiger Cub, $99)
220 Bolyston St., Suite 206
Chestnut Hill, MA 02167, USA
(Tel. 001/617/2446954)

Passport
(Master Tracks Pro)
625 Miramontes St.
Half Moon Bay, CA 94019, USA
(Tel. 001/415/7260280)

COMPUTERMUSICA A 8 BIT

LUNGO LA "VIA DELLA MUSICA" COMMODORE

Attrezzati di un Commodore 64/128, affrontiamo un viaggio musicale che ci porta da Mozart a Madonna... e oltre. Il tutto utilizzando il SID

di John Ryan

Con l'aumento della popolarità dei computer a 16 bit, quasi nessuno al di fuori degli utenti Commodore più affezionati si rende conto che il C-64/128 possiede un eccellente generatore di musica interno: il chip SID (Sound Interface Device). Nel 1982, quando il C-64 fece la sua comparsa sul mercato, questo circuito integrato era decisamente in anticipo sui tempi, e grazie a lui ancor oggi i computer Commodore a 8 bit sono tra le più efficienti "macchine musicali" esistenti. Per sfruttare il chip, però, occorre il software. E quali sono i programmi oggi disponibili a questo scopo? Se volete saperlo continuate a leggere e seguiteci lungo la "via della musica" Commodore.

Per cominciare

Music Construction Set (Electronic Arts) è stato il primo importante programma di elaborazione musicale per il C-64/128: un vero dono del cielo per i musicisti commodoriani. Pur riuscendo a sfruttare appieno alcune delle più avanzate caratteristiche del SID, *MCS* è semplice da usare, permette di eseguire elaborazioni in tempo reale e produce un feedback immediato.

Lo schermo di edit dà la possibilità di utilizzare pentagrammi in chiave di violino e di basso, con un puntatore controllato tramite mouse o tastiera e icone per selezionare le varie funzioni del programma. Per creare la musica si "prendono" note, diesis, bemolle, bequadri, legature e pause, e si collocano sulle chiavi spostandole tra le ottave come si preferisce. Si può far eseguire il brano in qualsiasi momento, e in seguito (se necessario) modificarlo tramite la funzione di cut & paste di *MCS*. Eventualmente si può anche "asportare" una misura da un altro brano e inserirla in quello su cui si sta lavorando.

Altre caratteristiche di *MCS* sono il controllo delle ottave e la selezione degli "strumenti". Il suono viene impostato scegliendo da 13 diverse configurazioni, otto delle quali imitano strumenti tradizionali come il flauto e il clavicembalo; le altre producono suoni percussivi o effetti speciali.

La versione a 16 bit di questo programma riscuote attualmente un vasto successo, ma la prima incarnazione di *Music Construction Set*, quella per le macchine a 8 bit, non raccoglie più la grande popolarità di un tempo. Imparare a usarlo è effettivamente molto semplice, ma i risultati in termini di qualità del suono non superano l'aurea mediocrità dal momento che il programma non fornisce un completo controllo delle capacità del SID. Inoltre sta diventando difficile trovare nuovi file musicali *MCS* sui BBS e sui vari network che in passato ne proponevano in abbondanza.

MCS resta comunque un ottimo programma per avviarsi lungo la via della musica Commodore. Soltanto più avanti, diventando compositori più smaliziati, se ne cominciano ad avvertire i limiti.

In studio

The Music Studio (Mediagenic) è un pacchetto più ambizioso. È composto da un gruppo di programmi che affrontano la composizione musicale secondo una logica "modulare", riuscendo a raggiungere risultati migliori di *MCS* sia come qualità sia come varietà. *The Music Studio* è anche MIDI-compatibile ed è in grado di effettuare la stampa degli spartiti.

La sezione principale di questo programma-multiplo è l'editor: come quello dell'*MCS* fornisce un doppio pentagramma e i comandi per la collocazione di note e pause; ma, a differenza di *MCS*, permette di accompagnare la notazione musicale con un testo (purché quest'ultimo non superi la lunghezza di quattro righe).

Le composizioni realizzate con *The Music Studio* si possono eseguire utilizzando 15 diversi strumenti. E se al compositore un suono non piace, può intervenire cambiando i valori di attacco, decadimento, sostentamento e rilassamento (ADSR), i valori delle forme d'onda e i vari filtri che alterano la risonanza, la sincronizza-

zione e la modulazione "ad anello" (si veda anche l'articolo "L'interfaccia sonora del C-64: il SID" pubblicato nel numero 4/88 di *Commodore Gazette*). Queste modifiche possono essere apportate nota per nota o sezione per sezione.

Se non siete musicisti provetti, apprezzerete in modo particolare un'innovativa tecnica di notazione musicale proposta da *The Music Studio*: il Paintbox. Tutto quello che si deve fare è "disegnare" rettangoli colorati lungo un rigo: la lunghezza di ogni rettangolo specifica il suono, mentre il colore indica lo strumento da utilizzare per l'esecuzione. Il Paintbox rappresenta una simpatica scorciatoia, e i pezzi scritti con questa tecnica possono sempre essere trasferiti nel normale editor del programma.

The Music Studio è uno strumento di composizione eccellente, che garantisce un controllo maggiore rispetto a *MCS* e permette di lavorare più in fretta e con maggiore efficienza. Anche la qualità del suono è migliore, e il Paintbox è un aiuto non indifferente per chi sia ancora inesperto nel campo della notazione. Il manuale, anch'esso di ottima qualità, è arricchito da una serie di indici che ne rendono particolarmente rapida la consultazione.

Musicisti con il SID

Al giorno d'oggi, il più popolare programma di composizione ed esecuzione musicale per C-64/128 è probabilmente *Enhanced Sidplayer*, di Craig Chamberlain. Secondo la mia personale opinione si tratta del più elaborato ed efficace programma musicale disponibile attualmente. *Sidplayer* viene distribuito su un disco a doppia faccia, con la versione per C-64 su un lato e quella per C-128 sull'altro; il disco è accompagnato da un manuale di 274 pagine rilegato a spirale.

Le caratteristiche di edit di questo programma forniscono un controllo totale del SID. Non soltanto permettono all'utente di scegliere tonalità, forma d'onda e inviluppo per ciascuna delle tre voci del C-64, ma anche di controllare i filtri, la modulazione, il vibrato, il portamento, la trasposizione e così via. Comunque, se preferite, il programma si occupa personalmente della regolazione del filtro (lo strumento che permette di modificare i timbri delle voci). *Enhanced Sidplayer* è così versatile che i file da esso generati si possono "fondere" con programmi Basic per essere usati come colonna sonora.

Come gli altri programmi di composizione, il *Sidplayer* offre un editor che permette di disporre la notazione musicale su un doppio pentagramma, eseguire operazioni di cut & paste e di ricerca, e ovviamente di suonare la musica composta. Ma *Sidplayer* porta un passo più avanti l'arte della composizione. L'eccellente documentazione allegata si spinge a fondo nel reame della modulazione, dell'impostazione dei filtri, della teoria musicale e così via, guidando l'utente passo per passo lungo il percorso della creazione musicale.

Un programmino autonomo di esecuzione, compreso nel pacchetto, permette di suonare l'enorme quantità di file *Sidplayer* di pubblico dominio che provengono da fonti come Q-Link e GEnie (due network americani, *ndr*). Vi si trova di tutto, dagli ultimi successi in classifica fino alla musica classica, da pezzi singoli a interi album, e molti di questi pezzi sono stupendi. Gli iscritti a Q-Link possono persino "suonare" i file *Sidplayer* direttamente dalla rete, grazie a un fantastico servizio chiamato "The Music Connection". Inoltre Q-Link possiede un attivissimo servizio d'informazioni a disposizione di tutti gli utenti.

In aggiunta ai file *Sidplayer* standard, sono stati realizzati anche file stereo. Però, dal momento che il C-64/128 non è in grado di produrre suoni stereofonici senza "aiuti" esterni, per apprezzare le sei voci fornite dall'esecuzione stereo ci si deve attrezzare di un apposito dispositivo hardware.

Il più semplice di questi dispositivi è la SID Symphony Stereo Cartridge, della Dr. Evil Laboratories. Collegata direttamente alla porta d'espansione, invia tre voci supplementari a un amplificatore esterno e a un altoparlante. La versione originale della cartuccia era alimentata da una pila alcalina a 9 volt collegata al sistema esterno tramite un semplice cavo RCA maschio/maschio. Ora c'è una nuova versione che dovrebbe essere alimentata direttamente dalla user port, ma non se ne conosce ancora il prezzo.

Dopo aver fatto esperimenti per tre mesi con questa piccola gemma, ora vi assicuro che non riesco più fare a meno dello stereo! I file *Sidplayer*, comunque, devono essere scritti specificamente per approfittare di questa capacità, altrimenti il risultato è comunque mono. La maggior parte dei file *Sidplayer* di pubblico dominio specificano chiaramente se sono stereo o mono.

Sono inoltre disponibili, sempre nelle librerie di pubblico dominio, decine di programmi di supporto per il *Sidplayer* e per la cartuccia, accompagnati dalla necessaria documentazione. Un esempio versatile ed efficiente è *Stereo V1.0* (QuantumLink, di Robert Stoerre, filename "Stereo.Arc"; si trova in vendita per 5 dollari anche presso la Dr. Evil Laboratories, che vende anche il sistema *Enhanced Sidplayer* per 22 dollari e 95 più spese di spedizione). Si tratta di una recente utility di "appoggio" all'editor dell'*Enhanced Sidplayer*, che permette d'intervenire contemporaneamente su tutte e sei le voci (con il solo *Sidplayer* si deve invece operare su una voce per volta). Fornisce inoltre capacità di edit e riproduzione MIDI, un editor per i titoli e i testi, e una semplicissima interfaccia a menu.

Non ci sono dubbi: *Sidplayer* è il miglior programma di composizione musicale per C-64/128, sia per i musicisti occasionali sia per i compositori seri. E se non avete intenzione di comporre musica, potete sempre usarlo per divertirvi ad ascoltare il lavoro dei computermusicisti di tutto il mondo.

Un facile ascolto

La Free Spirit Software ha creato una serie di trascrizioni musicali per il C-64/128 da "caricare e ascoltare" con l'unico ausilio del *Sidplayer*. La raccolta comprende una quindicina di dischi (tra cui i più noti sono *Music of the Masters* e *Strider's Computer Classics*) che contengono musiche di Bach, Mozart, Beethoven, Handel, Haydn, Joplin e altri compositori, accanto ai classici del gospel e alle più popolari canzoni natalizie. Alcuni brani non raggiungono la qualità dei file *Sidplayer* e *Music Studio*, ma ce ne sono parecchi di buona fattura.

Gli utenti di *GEOS* possono usufruire di due eccellenti editor musicali inclusi nel disco *GEOS Companion*, pubblicato dalla rivista *Run*: *G.O. Bach 40* e *G.O. Bach 80*, rispettivamente per C-64 e per C-128. Rapidi ed efficienti, questi editor presentano un'interfaccia utente progettata con insolita cura. Per di più, sul disco vi

sono alcune basi ritmiche preregistrate (accessibili dall'interno dei programmi) da usare per nuove composizioni musicali.

Rispetto a uno o due anni fa vi sono meno programmi musicali per gli utenti del C-64/128, ma quelli che abbiamo illustrato sono più che sufficienti per raggiungere in breve tempo eccellenti risultati, qualunque sia il vostro grado di preparazione musicale. Come primo esperimento potete provare a registrare qualche canzone dalla radio e a trasferirla nel formato del computer, oppure tentate di scrivere qualcosa voi stessi. Scoprirete che la via della musica Commodore è ricca di soddisfazioni e di ricompense. ■

SID: IL MAESTRO DI MUSICA

Il "Sound Interface Device", o SID, è il chip che sta alla base delle capacità musicali del C-64/128. A differenza degli strumenti tradizionali, dove si soffia attraverso un'imboccatura, si muove un archetto lungo una corda o si percuote una superficie, il SID crea i suoni digitalmente, attraverso innalzamenti e abbassamenti di tensione, bit accesi o spenti. Ma il risultato finale, quando gli impulsi elettronici vengono inviati a un altoparlante, è sempre lo stesso: vibrazioni nell'aria e nei padiglioni auricolari.

La frequenza di una vibrazione determina l'*altezza* del suono prodotto, mentre la *forma d'onda* determina la qualità (o timbro) del suono. È la forma d'onda che distingue il suono di un violino da quello di una chitarra, o un flauto da un pianoforte

Il chip SID contiene tre oscillatori che generano le forme d'onda delle tre voci del C-64/128. Sono disponibili quattro forme d'onda, con varie frequenze: triangolare, a rampa, quadra e aperiodica. Quella triangolare dà origine a un suono dolce, mentre quella a rampa produce un suono più frizzante; quella aperiodica è in grado di produrre qualsiasi suono, dai rombi più bassi ai sibili più acuti (viene utilizzata in genere per gli effetti sonori).

Il suono è definito ulteriormente dall'inviluppo, i cui parametri sono il tempo di attacco, il tempo di decadimento, il livello di sostentamento e il tempo di rilassamento (parametri ADSR). Per capire il significato di questi parametri pensate a un tuono: l'*attacco* è il tempo che il suono impiega a raggiungere il massimo volume (un tuono ha in genere un tempo di attacco brevissimo, visto che raggiunge il volume massimo quasi istantaneamente); il *decadimento* è il tempo che il suono impiega per scendere fino al livello di *sostentamento*, ovvero il livello in cui il suono si mantiene costante o diminuisce più lentamente (un tuono decade lentamente, svanendo a poco a poco in distanza). Il *rilassamento* è il tempo che impiega infine il volume per arrivare a zero.

Con il C-64, i programmatori che desiderano produrre un suono con il chip SID, devono specificare tutte queste caratteristiche per ogni vibrazione presente nelle tre voci, inserendo i valori negli opportuni registri tramite comandi POKE. Questo comprende i registri ADSR, di altezza e di volume, e inoltre a un bit che abilita l'esecuzione del suono. La produzione di effetti sonori costringe a ricorrere ad alcune delle caratteristiche più complesse del SID, come il filtro (che consente soltanto il passaggio di determinate frequenze), la modulazione ad anello, la sincronizzazione e la risonanza.

Nella maggioranza dei casi è importante anche l'ordine in cui vengono impartiti i comandi POKE. Per esempio, si deve azionare il controllo di volume e specificare una forma d'onda *prima* di abilitare il suono, altrimenti non si sente nulla. E nemmeno si possono inserire numeri a caso nei registri: il chip SID può produrre moltissimi suoni (per la precisione 65 536 in otto ottave), ma nei registri si devono inserire soltanto valori predefiniti oppure i risultati non sono prevedibili.

I programmatori del C-128, per loro fortuna, hanno a loro disposizione un Basic eccellente, che elimina il complesso lavoro di assegnare i valori ai registri tramite i comandi POKE. Con il Basic 7.0 tutto ciò che il programmatore deve fare è specificare i valori tramite i comandi Vol, Sound, Play, Envelope, Tempo e Filter. Per esempio, l'istruzione SOUND 3,49000,200,1,0,100,1,0 equivale a parecchie linee scritte nel Basic del C-64

Il Sound Interface Device è un chip di supporto meravigliosamente complicato, in grado di produrre suoni che vanno dai colpi di laser alla musica classica Per programmare questo chip è d'obbligo conoscere discretamente la teoria musicale, l'architettura del SID e, naturalmente, la programmazione. Se non possedete ancora queste conoscenze, potete sempre passare qualche serata ad ascoltare musiche scritte da compositori arrivati a un livello un po' più avanzato del vostro

Per ulteriori informazioni contattare direttamente:

Compute! Publications, Inc.
(Enhanced Sidplayer, $24.95)
P.O. Box 5406
Greensboro, NC 27403
USA

Dr. Evil Laboratories
(The SID Symphony
Stereo Cartridge, $34.95)
P.O. Box 3432
Redmond, WA 98073-3432
USA

Electronic Arts
(Music Construction Set, $14.95)
1820 Gateway Drive
San Mateo, CA 94404
USA

Free Spirit Software
(Music of the Masters,
Strider's Computer Classics, $5 l'uno)
P.O. Box 128
Kutztown, PA 19530
USA

Mediagenic
(The Music Studio, fuori catalogo)
3885 Bohannon Drive
Menlo Park, CA 94025
USA

QuantumLink Computer Service
(QuantumLink)
8620 Westwood Center Drive
Vienna, VA 22180
USA

Run Magazine
(GEOS Companion, $24.97)
80 Elm Street
Peterborough, NH 03458
USA

ABBONATI!

A CASA TUA UN REGALO OGNI MESE

• Possibilità d'includere nell'abbonamento gli arretrati
• Comodità di ricevere la rivista direttamente a casa

Per chiunque sia interessato ai computer Commodore, *Commodore Gazette* è indispensabile. Nessuna rivista in Italia offre ai suoi lettori tanta qualità con recensioni hardware e software, listati, presentazioni esclusive, informazioni di ogni genere riguardanti C-64, C-128 e Amiga.

Un abbonamento a *Commodore Gazette* è il regalo più bello che possiate fare a voi stessi e agli altri... un regalo nuovo ogni mese. Ma non è finita! Risparmierete il 15% sul prezzo di copertina e potrete includere nel prezzo dell'abbonamento anche i numeri arretrati che mancano alla vostra raccolta. Resta inteso che per ogni arretrato scelto verrà spedito un numero di prossima uscita in meno. Per esempio, chi si abbona a 12 numeri a partire dal n. 1/89 e richiede 5 arretrati, riceverà 7 numeri del 1989 e 5 arretrati.

Ritagliare e spedire a: IHT Gruppo Editoriale – Via Monte Napoleone, 9 – 20121 Milano

Sì, desidero sottoscrivere un abbonamento a ☐ 12 numeri (L. 81.600) ☐ 24 numeri (L. 163.200) di COMMODORE GAZETTE usufruendo così dello sconto del 15% sul prezzo di copertina

Nome e cognome ....
Indirizzo .... Città .... CAP ....
Inizio abbonamento dal n. .... Desidero ricevere i seguenti numeri arretrati (specificare numero e anno) ....
☐ Allego assegno bancario o circolare ☐ Allego fotocopia della ricevuta del vaglia postale

Firma ....

PROGRAMMARE L'AMIGA

LE FUNZIONI DOS PER LA GESTIONE DI FILE E DIRECTORY

Una trattazione "teorica" di alcune funzioni della libreria DOS, con la descrizione dei parametri richiesti e dei valori restituiti

di Eugene P. Mortimore

Quest'articolo pone le basi per un'approfondita discussione sulle funzioni della libreria ARP. Nel precedente articolo della serie abbiamo presentato sommariamente tutte le 103 funzioni della libreria, separandole in due tavole.

Ora discuteremo in maggior dettaglio le funzioni DOS classificabili come funzioni per la gestione di file e directory. Si ricordi, comunque, che la libreria DOS include anche cinque funzioni classificabili come funzioni per la gestione dei processi (CreateProc, DateStamp, Delay, DeviceProc ed Exit), tre funzioni classificabili come funzioni per la manipolazione dei codici (Execute, LoadSeg e UnLoadSeg) e due funzioni classificabili come funzioni relative ai pacchetti DOS (GetPacket e QueuePacket). Ne parleremo in uno dei prossimi articoli.

Ci siamo concentrati sulle funzioni per la manipolazione di file e directory perché su di esse si basano molte delle nuove funzioni ARP. In secondo luogo, quasi tutti i programmi che fanno uso delle funzioni ARP ricorrono anche alle funzioni DOS. Per muoversi tra le directory, per esempio, si devono chiamare le funzioni CreateDir, CurrentDir e/o ParentDir, e in seguito le funzioni Examine e ExNext. La libreria ARP non fornisce funzioni nuove e migliori che le sostituiscano.

Le funzioni DOS per file e directory

La Tavola 1 mostra le 23 funzioni DOS per la gestione di file e directory, ovvero le funzioni che hanno a che fare con il flusso di I/O tra il programma e le finestre CLI. La tavola illustra i principali aspetti e le interazioni reciproche delle funzioni, concentrandosi in particolare sul loro impiego, sul significato dei loro argomenti e sui valori da esse restituiti. Come si vede, le funzioni sono raggruppate in dieci categorie. Prima di procedere oltre, però, vediamo quali osservazioni generali si possono fare sulla libreria DOS.

Tutte le funzioni della libreria DOS collocano automaticamente i loro valori di ritorno (se ve ne sono) nel registro D0. Si noti che le funzioni che non restituiscono alcun valore possono comunque alterare qualche parametro di struttura.

Tutte le funzioni della libreria DOS, inoltre, si aspettano di trovare i loro argomenti nei registri D1, D2 e D3 della CPU. Il primo argomento viene sempre collocato nel registro D1, il secondo nel registro D2 e il terzo nel registro D3. Di questa caratteristica va tenuto conto quando si scrive un programma in Assembly (se il programma è in C, se ne occupa direttamente il compilatore). Si noti che queste assegnazioni di registri sono leggermente diverse rispetto a quelle di altre funzioni del ROM Kernel, in quanto tutti gli argomenti vengono collocati nei registri-dati della CPU, anche quando si tratta di valori "indirizzati", come nel caso di puntatori a stringhe a terminazione nulla. Un'ultima osservazione: ricordate che le funzioni C standard per la gestione di file e directory sono state progettate *sopra* le originali funzioni della libreria DOS. In altre parole, i programmatori della Manx e della Lattice hanno costruito tutte le loro funzioni per la gestione di file e directory (fopen, fclose...) ricorrendo alle funzioni che ci apprestiamo a descrivere.

Per questo motivo, se vi basate soltanto sulle funzioni della libreria DOS il vostro programma sarà in generale *più veloce* che se usaste le funzioni delle librerie C standard. Di solito, per comodità, i programmi sono una miscela di queste due classi di funzioni.

Le funzioni Input e Output. Le prime due funzioni della tavola sono Input e Output: entrambe non richiedono argomenti, e restituiscono un puntatore a una struttura FileHandle.

Nel sistema multitasking dell'Amiga è possibile mandare in esecuzione un programma direttamente da una finestra CLI come processo in *foreground* (come si fa con i task) e successivamente "perdere contatto" con la finestra CLI finché il programma stesso non restituisce

continua a pagina 60

Le funzioni della libreria DOS/ARP per la gestione di file e directory

Nome	Valore restituito (sempre in D0) (0 significa errore; le eccezioni sono segnalate)	Argomenti	Registri
Input	fileHandle	Nessuno	Nessuno
Output	fileHandle	Nessuno	Nessuno
Open	fileHandle	fileNale, access_mode	D1, D2
Open	fileHandle	"*:", MODE_OLDFILE	D1, D2
Open	fileHandle	"CON.10/10/400/100/New Window", MODE_NEWFILE	D1, D2
Open	fileHandle	"RAW.10/10/400/100/Raw Window", MODE_NEWFILE	D1, D2
Open	fileHandle	"PRT.", MODE_NEWFILE	D1 D2
Open	fileHandle	"SER.", MODE_NEWFILE	D1, D2
Open	fileHandle	"PAR:", MODE_NEWFILE	D1 D2
Close	Nessuno	fileHandle	D1
Read	actual_length(*)	fileHandle, fileBuffer, desired_length	D1, D2 D3
Write	actual_length(*)	fileHandle, fileBuffer, desired_length	D1 D2 D3
Seek	old_position(*)	fileHandle, desired_position, access_mode	D1, D2 D3
Rename	success	oldFileName, newFileName	D1, D2
Rename	success	oldDirName, newDirName	D1 D2
DeleteFile	success	fileName	D1
DeleteFile	success	dirName	D1
CreateDir	dirLock	dirName	D1
CurrentDir	oldDirLock(**)	fileLock	D1
CurrentDir	oldDirLock(**)	dirLock	D1
ParentDir	parentDirLock(**)	fileLock	D1
ParentDir	parentDirLock(**)	dirLock	D1
Examine	success	fileLock, fileInfoBlock	D1 D2
Examine	success	dirLock, fileInfoBlock	D1 D2
ExNext	success	fileLock, fileInfoBlock	D1 D2
ExNext	success	dirLock, fileInfoBlock	D1 D2
Lock	fileLock	fileName, lock_mode	D1 D2
Lock	dirLock	dirName lock_mode	D1 D2
DupLock	newFileLock	oldFileLock	D1
DupLock	newDirLock	oldDirLock	D1
UnLock	Nessuno	fileLock	D1
UnLock	Nessuno	dirLock	D1
Info	success	fileLock, infoData	D1 D2
Info	success	dirLock infoData	D1 D2
SetComment	success	fileName fileComment	D1 D2
SetComment	success	dirName, dirComment	D1 D2
SetProtection	success	fileName, protection_mask	D1 D2
SetProtection	success	dirName protection_mask	D1 D2
IsInteractive	success	fileHandle	D1
WaitForChar	success	fileHandle time_out	D1 D2
IoErr	error_number	Nessuno	Nessuno

(*) La condizione d errore è indicata dal valore 1 (e non 0)
(**) Il valore 0 indica che è stata raggiunta la directory radice

il controllo tramite un'istruzione exit; oppure si può mandare in esecuzione il programma come processo in *background*, utilizzando i comandi RUN o EXECUTE dell'AmigaDOS, permettendo così all'utente d'impartire altri input dalla finestra CLI di partenza. Una parte di questi input (o anche tutti) possono essere diretti al programma. Ora, ogni flusso di I/O nel sistema Amiga è associato a una struttura FileHandle. A volte queste strutture vengono allocate e inizializzate automaticamente dalle routine interne dell'AmigaDOS, ma altre volte se ne occupa esplicitamente il programma per soddisfare le sue necessità di dati. Per esempio, ogni volta che aprite una finestra CLI utilizzando il comando NEWCLI, le routine interne dell'AmigaDOS allocano e inizializzano automaticamente una struttura FileHandle per caratterizzare il flusso I/O di caratteri con quella particolare CLI. L'AmigaDOS utilizza internamente queste strutture FileHandle e i loro parametri variabili, ma anche i programmi, come ora vedremo, le possono sfruttare per i loro scopi.

La funzione Input restituisce un puntatore alla struttura FileHandle che "appartiene" al flusso di input tra tastiera e finestra CLI.

Per esempio, se avete un file eseguibile denominato MioProgramma e lo mandate in esecuzione dalla finestra CLI numero 1, all'interno di MioProgramma potete inserire una chiamata a Input: Input restituirà il puntatore alla struttura FileHandle che caratterizza il flusso di input della finestra numero 1.

Questo meccanismo di input è ovviamente utile se il programma ha in qualche modo la necessità di collegarsi con l'input inviato da CLI. La procedura è semplicissima: si chiama la funzione Read indicando come primo argomento il puntatore alla struttura FileHandle restituito dalla funzione Input.

Allo stesso modo, la funzione Output ha lo scopo di procurare al programma un puntatore alla struttura FileHandle che appartiene al flusso di output tra la finestra CLI e lo schermo. Questo è utile soltanto se il programma intende inviare un testo in output sulla finestra CLI di partenza; utilizzando infatti come primo argomento il puntatore alla struttura FileHandle restituito da Output, il programma può chiamare la funzione Write e inviare un messaggio all'utente, ad esempio per segnalare un errore di sintassi nel testo digitato in linea comando.

Le funzioni Open e Close. La prima serve per aprire un flusso di dati (in genere un file su disco), e trasmettere o ricevere caratteri.

Open richiede come primo argomento una stringa a terminazione nulla con il nome del file, e come secondo argomento il "modo di accesso" al flusso di dati. Se per esempio voleste aprire un *nuovo* file denominato MioFile, dovreste seguire questa sintassi:

```
struct FileHandle *mioFileHandle,
mioFileHandle = Open(MioFile  MODE_NEWFILE);
```

Quando Open restituisce il controllo, mioFileHandle punta a una struttura FileHandle che rappresenta il *nuovo* flusso di dati. Open riconosce i seguenti modi d'accesso: MODE_OLDFILE (per aprire un file già esistente) e MODE_NEWFILE (per aprire un nuovo file). Se la funzione non può aprire il file (per esempio perché non esiste nessun file "MioFile" nel percorso di ricerca indicato) Open restituisce il valore 0.

La tavola a pagina 59 elenca anche altri "tipi di file" che si possono aprire con Open.

Prima di tutto si può aprire la finestra di lavoro, la finestra da cui il programma è stato mandato in esecuzione, utilizzando la notazione "*:". Utilizzando invece la notazione "CON:", si può aprire e collocare in una posizione specifica dello schermo una finestra il cui input viene preelaborato dalle routine interne del dispositivo Console. Con la notazione "RAW:" si apre invece una finestra collegata col dispositivo Console, i cui dati restano però grezzi. La scelta dell'uno o dell'altro tipo di finestra influenza il modo in cui appaiono sullo schermo i caratteri digitati.

Tramite la notazione "PRT:" si può poi aprire un file del dispositivo Printer. Serve per inviare alla stampante un file su disco, agendo dall'interno del programma: prima di tutto si chiama la funzione Open (filename "PRT:" e modo d'accesso MODE_NEWFILE) per ottenere un puntatore alla struttura FileHandle per un nuovo "file stampante", poi si chiama ancora Open per aprire un file su disco già esistente (MODE_OLDFILE) ottenendo il puntatore alla struttura FileHandle di quel file. A questo punto si chiama la funzione Read (indicando quest'ultimo puntatore) per leggere il testo del file su disco e collocarlo in un RAM buffer del programma. Infine si chiama la funzione Write (stavolta indicando il primo dei due puntatori) per trasferire il testo dal RAM buffer al file stampante.

Questo tipo di sequenza può essere usata anche per inviare dati alla porta seriale (notazione "SER:") o alla porta parallela (notazione "PAR:"): la funzione Open può essere impiegata per spostare in ogni locazione del sistema i dati immagazzinati su disco.

La funzione Close serve per chiudere un "flusso di dati" precedentemente aperto, in genere un file su disco. Come unico argomento, questa funzione richiede un puntatore a una struttura FileHandle, restituito in precedenza da una chiamata a Open. Se per esempio volete chiudere il file che aveva mioFileHandle come puntatore, si deve effettuare la seguente chiamata:

```
struct FileHandle *mioFileHandle,
Close(mioFileHandle);
```

Quando Close restituisce il controllo, il flusso di dati indicato non è più accessibile dall'interno del programma; si tenga presente che Close è concepito per chiudere soltanto flussi di dati aperti tramite la funzione Open nel corso dello stesso programma.

Le funzioni Read e Write. Una volta che il file è stato aperto, si possono leggere i suoi dati con la funzione Read o se ne possono scrivere di nuovi con la funzione Write. Come si vede dalla tavola, Read richiede tre argomenti: fileHandle (puntatore alla relativa struttura FileHandle), fileBuffer (puntatore al buffer dove verranno immagazzinati i caratteri contenuti nel file) e desired_length (il numero di caratteri che si vogliono leggere dal file).

Se la chiamata a Read ha successo, nella variabile actual_length viene restituito l'*effettivo* numero di caratteri letti. Se Read restituisce 0 significa che è stata raggiunta la fine del file. Se infine Read s'imbatte in qualche tipo di errore, restituisce il valore −1. Per saperne di più sull'errore, il programma chiama come sempre IoErr.

Oltre a controllare se il valore restituito è 0 o −1,

conviene anche confrontare il numero di caratteri richiesto con quello effettivamente letto: se ci sono differenze significa che si è verificato qualche imprevisto. Tenete sempre presente che ogni "lettura" di un file comincia dall'ultima posizione impostata tramite una chiamata alla funzione Seek (vedere più avanti).

La funzione Write preleva il testo contenuto in un buffer di programma e lo trasferisce in un file precedentemente aperto. I suoi argomenti sono identici a quelli della funzione Read e hanno significati analoghi. Per la precisione, il secondo argomento (fileBuffer) punta a un buffer in cui il programma ha collocato in precedenza dei caratteri (non ha importanza in che modo). Per copiare ad esempio il contenuto di un file in un altro file di nome diverso, si possono aprire entrambi i file con Open (ottenendo i puntatori alle relative strutture FileHandle), poi "leggere" tramite Read il contenuto del primo file in un buffer di opportuna grandezza e infine trasferirlo nell'altro file tramite la funzione Write.

Una volta afferrati questi concetti, è piuttosto semplice scriversi da soli i codici di un comando Copy che operi allo stesso modo del comando COPY dell'AmigaDOS. Basta che il puntatore fileBuffer sia lo stesso per le due chiamate a Read e a Write, e che i puntatori fileHandle puntino a due diverse strutture FileHandle.

Ma si può utilizzare Write anche per trasferire in un nuovo file un testo collocato in un buffer in altro modo, per esempio stringhe di testo dichiarate come variabili globali. Si deve semplicemente aprire il nuovo file tramite Open, e indicare nella chiamata a Write il puntatore al buffer che contiene il testo da trasferire.

Come con Read, conviene confrontare il numero di caratteri richiesto (indicato nella variabile desired_length) con quello effettivamente trasferito (indicato nella variabile actual_length), dopo che Write ha restituito il controllo.

La funzione Seek. Come si vede dalla tavola, la funzione Seek richiede tre argomenti: fileHandle (puntatore a una struttura FileHandle), desired_position (la posizione desiderata all'interno del file), e file_position_mode (il "modo di posizionamento" del file). Seek riconosce tre modi di posizionamento: OFFSET_BEGINNING, OFFSET_CURRENT e OFFSET_END. Se per esempio volete far avanzare di dieci byte l'immaginario cursore che indica la posizione raggiunta nel file, si deve impostare a 10 l'argomento desired_position e file_position_mode a OFFSET_CURRENT. Seek restituisce sempre la posizione raggiunta dal cursore. In particolare, se si chiama Seek indicando 0 nell'argomento desired_position e OFFSET_CURRENT come modo di posizionamento, viene semplicemente restituita la posizione del cursore. Per collocare il cursore alla fine del file basta chiamare Seek con desired_position pari a 0 e con OFFSET_END come modo di posizionamento.

Le funzioni Rename e DeleteFile. Entrambe restituiscono un semplice valore booleano (ovvero TRUE o FALSE). I valori numerici di queste costanti sono 1 per TRUE e 0 per FALSE.

Se il cambiamento di nome effettuato con Rename ha avuto successo viene restituito il valore 1. Gli argomenti della chiamata sono due: oldName (puntatore a una stringa a terminazione nulla che rappresenta il vecchio nome del file) e newName (puntatore a una stringa a terminazione nulla che rappresenta il nuovo nome del file). Si noti che Rename non fa riferimento alla struttura FileHandle del file: per cambiar nome a un file è sufficiente chiamare Rename, e poi chiamare Open indicando il nuovo nome come primo argomento.

OldName e newName possono essere anche complessi tracciati di ricerca come per esempio DF1:dir1/MioFile1 e DF1:dir2/MioFile2. Si noti che questo particolare cambiamento di nome ha l'effetto di *spostare* il file da DF1:dir1 a DF1:dir2, e cambiandone contemporaneamente il nome da MioFile1 a MioFile2. Se cioè si cambia il percorso di ricerca cambiando directory, si ottiene l'effetto di *spostare* effettivamente il file (non copiarlo) nella seconda directory. Ricordate però che la seconda directory deve già esistere, altrimenti viene restituita una condizione d'errore.

Attenzione: è impossibile rinominare un file da un disco all'altro. In altre parole, il nome DF0:dir1/MyFile1 non può diventare DF1:dir2/MyFile2, perché equivarrebbe a spostare un file da un dispositivo all'altro e il sistema non permette un'operazione simile.

A parte il cambiamento di nome da un dispositivo all'altro, si può rinominare (e spostare effettivamente) qualunque file in qualunque directory del sistema. Se esiste già un file con il nuovo nome (nella nuova directory), Rename restituisce una condizione d'errore (FALSE). Si determina il tipo d'errore chiamando, come sempre, IoErr.

DeleteFile richiede un solo argomento: una stringa a terminazione nulla che rappresenta il nome del file. Si noti che, come avviene con Rename, i file vengono cancellati riferendosi soltanto al loro nome, non alla loro struttura FileHandle. Sottolineamo, quindi, che i file *non hanno bisogno di essere aperti* per essere rinominati o cancellati. Le due funzioni possono inoltre agire su intere directory.

Le funzioni CreateDir, CurrentDir e ParentDir. CreateDir ha lo scopo di creare una nuova directory; richiede un solo argomento: una stringa a terminazione nulla che rappresenta il nome di una directory.

L'AmigaDOS richiede che ogni directory sia associata a una struttura FileLock: se CreateDir si conclude con successo, restituisce un puntatore (directoryLock) alla struttura FileLock della directory.

La funzione CurrentDir richiede come unico argomento un puntatore alla struttura FileLock della directory che deve diventare quella di lavoro.

Ogni volta che si indica un file senza specificare il nome della directory, il sistema lo cerca nella directory di lavoro. In particolare, ogni file aperto dal programma deve trovarsi nella directory di lavoro, a meno che nella chiamata alla funzione Open non sia specificamente indicato anche il nome della directory.

Se non si verificano errori, CurrentDir restituisce un puntatore (oldFileLock) alla struttura FileLock associata alla *vecchia* directory di lavoro. Questo valore può essere salvato in una variabile locale, nel caso che si desideri tornare in seguito alla vecchia directory tramite un'altra chiamata a CurrentDir. In particolare, si deve chiamare CurrentDir nel momento in cui si entra nel programma: salvando in una variabile locale il puntatore che viene restituito e usandolo per un'altra chiamata a CurrentDir prima dell'uscita, ci si assicura che l'uscita avvenga nella finestra CLI da cui il programma era stato mandato in esecuzione.

La funzione ParentDir assomiglia a CurrentDir, dal momento che cambia la directory di lavoro, ma è più

specifica: serve a far diventare directory di lavoro la directory genitore, ovvero quella immediatamente precedente nell'albero delle directory. Chiamando ripetutamente questa funzione si arriva alla directory radice.

ParentDir richiede come unico argomento il puntatore alla struttura FileLock di un file o di una directory, e restituisce il puntatore (parentFileLock) alla struttura FileLock della directory o del file genitore. Se il file di lavoro è inferiore soltanto di un livello rispetto alla radice (cioè se il genitore è la directory radice) viene restituito il valore 0.

Le funzioni Examine ed ExNext. A volte un programma ha bisogno di sapere quali sono i file di una directory o di una sotto-directory, e se si possono visualizzare e correggere. A questo scopo, le funzioni Examine ed ExNext forniscono un meccanismo per spostarsi tra i file di una specifica directory e delle sue sotto-directory.

Prima di tutto si deve chiamare CurrentDir o ParentDir per trasformare la directory che c'interessa in quella di lavoro. Poi il programma può chiamare Examine, ed effettuare quindi una serie di chiamate a ExNext per avere informazioni su tutti i file della directory. Entrambe le funzioni restituiscono un valore booleano, ed entrambe richiedono due argomenti: fileLock o directoryLock (puntatore a una struttura FileLock o DirectoryLock) e fileInfoBlock (puntatore a una struttura FileInfoBlock). Examine impiega le informazioni che trova nella struttura FileLock per inizializzare la struttura FileInfoBlock; queste informazioni comprendono il nome, la data di creazione e la natura dell'ente associato alla struttura (file o directory). In precedenza, però, il programma deve aver allocato la struttura FileInfoBlock (si osservi che questa struttura deve essere allineata alle long word in RAM; se si usa la funzione AllocMem dell'Exec, l'allineamento è automaticamente rispettato).

Se Examine non riesce a inizializzare correttamente la struttura FileInfoBlock, restituisce uno 0.

ExNext serve invece per esaminare la voce successiva di una directory. Le operazioni da svolgere per esaminare i file di una directory sono le seguenti:

(1) si chiama la funzione Examine utilizzando il puntatore alla struttura FileLock restituito dalle funzioni CreateDir o Lock. Examine inizializzerà parzialmente la struttura FileInfoBlock che in precedenza dev'essere già stata allocata e associata alla directory.

(2) Si chiama la funzione ExNext utilizzando lo stesso puntatore alla struttura FileLock e un puntatore alla struttura FileInfoBlock inizializzata da Examine.

(3) Si chiama ripetutamente ExNext finché non restituisce il valore 0. Si chiama infine IoErr per assicurarsi di aver raggiunto la fine della directory (il codice d'errore dev'essere ERROR_NO_MORE_ENTRIES).

Osserviamo, per inciso, che per sapere se una struttura FileInfoBlock si riferisce a un file o una directory basta controllare il valore del suo parametro fib_EntryType: se è negativo si tratta di un file, altrimenti di una directory. Solo in quest'ultimo caso ha senso chiamare ExNext.

Le funzioni Lock, DupLock e UnLock. Le tre funzioni successive della tavola hanno a che fare con i *lock* di file e directory. Il termine "lock" (una scelta non troppo felice) si riferisce alla capacità del sistema di "chiuder fuori" specifici programmi (processi) dall'accesso a particolari file o directory.

I lock sono di due tipi: lock di lettura (tutti i task o processi cooperanti possono leggere contemporaneamente quel file), e lock di scrittura (soltanto il task o processo in esecuzione può scrivere nel file).

Un programma può utilizzare la funzione Lock se deve determinare l'esistenza di un file o di una directory. Lock richiede due argomenti: il primo è il nome del file o della directory di cui si vuole ottenere il lock (ovvero una stringa a terminazione nulla), il secondo è il modo di accesso che si desidera per il file (ACCESS_READ o ACCESS_WRITE). Se tutto va bene, Lock restituisce il puntatore alla struttura FileLock associata al lock del file o della directory. Altrimenti restituisce uno 0. Tenete presente che il microprocessore risulta meno impegnato da una chiamata a Lock che da una chiamata a Open. Conviene quindi chiamare Lock, se si vuole soltanto accertare l'esistenza di un file.

DupLock ha invece lo scopo di fornire il puntatore a un'altra struttura FileLock *identica* a quella ottenuta da una chiamata alla funzione Lock o a CreateDir in un altro programma. In genere ciò accade quando due task hanno l'esigenza di leggere lo stesso file; il primo, ricorrendo all'appropriato filename, chiama Open e apre il file ottenendo un puntatore alla sua struttura FileHandle, cosa che gli permette di scrivere e leggere dati al suo interno (tramite le funzioni Write e Read). Sempre il primo programma chiama poi Lock (indicando lo stesso filename e il modo ACCESS_READ) e ottiene un puntatore alla struttura FileLock del file. A questo punto, il secondo task (che opera in multitasking con il primo) desidera leggere i dati dello stesso file; se chiama DupLock utilizzando il puntatore alla struttura FileLock restituito dal primo programma, ottiene il puntatore a una nuova struttura FileLock che può impiegare per riferirsi al file. Chiamando poi la funzione Open con lo stesso filename, il secondo task ottiene un puntatore alla struttura FileHandle del file e può chiamare a sua volta la funzione Read per leggere i dati che gli interessano.

DupLock richiede come unico argomento il puntatore a una struttura FileLock, e restituisce il puntatore a un'altra struttura FileLock relativa allo stesso file o directory. Si osservi che i lock in scrittura non si possono duplicare.

La funzione UnLock toglie i lock impostati dalle funzioni Lock, DupLock o CreateDir. Al contrario di quanto accade con le funzioni appena citate (che restituiscono un puntatore alla struttura FileLock), UnLock non restituisce nulla che il programma possa controllare. Il lock viene rimosso soltanto dal particolare task in cui UnLock è andato in esecuzione.

Le funzioni Info, SetComment e SetProtection. La funzione Info raccoglie informazioni sul contenuto dei file presenti nel disco e le trasmette al programma. L'output di Info è simile a quello del comando INFO dell'AmigaDOS, che viene mandato in esecuzione dalla linea comando.

Info richiede due argomenti: fileLock (puntatore a una struttura FileLock associata a una directory o a un file) e infoData (puntatore a una struttura InfoData). La struttura InfoData dev'essere allineata alle long word; per questo motivo conviene allocarla (come la struttura FileInfoBlock) tramite la funzione AllocMem dell'Exec.

Info inizializza alcuni parametri della struttura Info-Data, e restituisce un valore booleano per indicare il successo o il fallimento della chiamata.

SetComment inserisce un commento nella struttura FileInfoBlock di un file o di una directory (più precisamente nel parametro fib_Comment). Richiede come argomenti due puntatori a stringhe a terminazione nulla: la prima contiene il nome del file o della directory a cui si vuole aggiungere il commento, e la seconda contiene il commento stesso (limitato a 80 caratteri). Per indicare il successo o il fallimento della funzione, SetComment restituisce un valore booleano.

SetProtection permette al programma d'impostare i bit di protezione per un file o per una directory. Richiede due argomenti: il puntatore a una stringa a terminazione nulla che contiene il nome del file o della directory, e una *maschera di protezione* (ovvero un insieme di bit a 0 e a 1 in un long integer). I nomi simbolici di questi bit sono definiti nel file INCLUDE dos.h.

Come si vede anche nella tavola, la maschera di protezione è contenuta nel registro D2, che è un registro a 32 bit; gli unici bit che vengono utilizzati, comunque, sono i quattro più bassi. Ecco il loro significato:

- Bit 3 impostato a 1: non sono permesse operazioni di LETTURA
- Bit 2 impostato a 1: non sono permesse operazioni di SCRITTURA
- Bit 1 impostato a 1: non è permessa l'ESECUZIONE del file
- Bit 0 impostato a 1: non è permessa la CANCELLAZIONE del file

Le funzioni IsInteractive e WaitForChar. IsInteractive permette ai programmi di sapere se un file (in senso lato) associato a una particolare struttura FileHandle è collegato a un terminale virtuale. Un terminale virtuale è ciò che permette all'utente d'inserire caratteri da tastiera e di visualizzarli su una finestra CLI. In questo contesto, le parole "terminale virtuale" e "interattivo" sono sinonimi.

WaitForChar fa in modo che il programma sappia che sta per arrivare un carattere da tastiera (un carattere digitato interattivamente dall'utente in un terminale virtuale CLI), che dev'essere inserito in un buffer di testo.

Come abbiamo accennato, nel linguaggio dell'Amiga una finestra CLI è un terminale virtuale; tramite la funzione Input ci si può procurare un puntatore alla struttura FileHandle associata al flusso di input di ogni particolare finestra. Da questo punto di vista si può dire che una macchina "fisica" come l'Amiga ha innumerevoli terminali "virtuali", ciascuno rappresentato da una finestra CLI.

IsInteractive restituisce un valore booleano che indica se effettivamente il flusso di I/O associato con la struttura FileHandle appartiene a un terminale virtuale.

WaitForChar è strettamente correlata alla funzione IsInteractive, e serve esclusivamente per i flussi di I/O relativi a terminali virtuali (finestre CLI). Prima di WaitForChar, quindi, si deve sempre chiamare IsInteractive, che dice al programma se la struttura FileHandle appartiene a un terminale virtuale oppure, per esempio, a un file su disco.

WaitForChar permette al programma d'individuare un singolo carattere digitato dall'utente in un terminale virtuale entro un certo tempo limite. Per esempio, si può far comparire sullo schermo un prompt che richieda all'utente di rispondere sì o no (S/N), facendo in modo che se l'utente non risponde entro un secondo il programma prosegua come se la risposta fosse sì.

Questa funzione richiede due argomenti: il puntatore alla struttura FileHandle del terminale virtuale e il numero di microsecondi d'intervallo. L'intervallo indica per quanto tempo il programma deve attendere che l'utente digiti un carattere. Anche WaitForChar restituisce un valore booleano: TRUE, se l'utente digita un carattere entro il tempo limite, FALSE in caso contrario. Se il risultato è TRUE, il programma chiama Read per leggere (cioè trasferire) il carattere dalla finestra a un buffer di testo. Se si desidera immagazzinare nel buffer più di un carattere, conviene collocare le chiamate a WaitForChar e Read in un ciclo "for" o "while", e costruire il buffer di testo carattere per carattere facendo avanzare in modo opportuno il "cursore" (o indice) del buffer.

La funzione IoErr. Dalla tavola si vede che tutte le funzioni che restituiscono un valore (soltanto due non lo fanno: Close e UnLock) in genere restituiscono 0 per indicare una condizione d'errore. Ci sono soltanto tre eccezioni a questa regola: Read, Write e Seek, che indicano la condizione d'errrore restituendo il valore –1. Questi due numeri, 0 e –1, sono chiamati "valori d'errore primari": dicono al programma che si è verificato un errore ma non forniscono nessuna indicazione sulla sua natura.

IoErr ha appunto lo scopo di mettere i programmi in condizione di determinare l'esatta ragione del fallimento di una chiamata. Essa restituisce un valore che viene chiamato "valore d'errore secondario" (si ricordi, però, che la chiamata dev'essere effettuata *immediatamente dopo* il verificarsi dell'errore). IoErr non richiede argomenti e restituisce un valore long integer (error_number) definito nel file INCLUDE dos.h. Consultando il file dos.h si scoprirà che esistono 31 #defines di valori d'errore, ai cui nomi simbolici sono associati numeri che vanno da 103L a 232L.

Facciamo qualche esempio: la funzione Write può segnalare un errore perché il disco su cui si tenta di salvare un file è già troppo pieno. In questo caso, IoErr restituisce il valore ERROR_DISK_FULL (valore numerico 221L). Se invece il disco non è pieno, ma è protetto in scrittura, IoErr restituisce il valore ERROR_WRITE_PROTECTED (223L). Il programma può predisporre opportune contromisure per queste due situazioni, che sono tra le più comuni, spiegando all'utente quali sono i provvedimenti da prendere (per esempio, spostare la linguetta di protezione, o cambiare disco).

Vi consigliamo di studiare il programma d'esempio presentato in appendice all'articolo "La libreria DOS/ARP al microscopio: prima parte", per vedere come vengono gestiti in pratica gli errori primari e secondari.

Nel prossimo numero

Per completare la trattazione iniziata con questo articolo, nel prossimo tratteremo in dettaglio le strutture di cui fanno uso le funzioni della libreria DOS: quelle già nominate qui (FileHandle, FileLock, FileInfoBlock e InfoData) e tutte le altre.

■

UNO SGUARDO SUGLI USA

TANTE MOSTRE E NOVITÀ, MA L'HARD DISK...

Si avvicina la fine degli 8 bit e delle riviste dedicate a questi computer. Il futuro dell'Amiga sembra invece pieno di promesse

di Morton A. Kevelson

Negli Stati Uniti il futuro dei computer a 8 bit della Commodore comincia ad apparire piuttosto grigio. L'ultima volta ho accennato al fatto che la rivista *Computer Shopper* aveva deciso di non occuparsi più dei cosiddetti "computer classici", tra cui C-64 e C-128. Adesso anche la rivista *INFO* ha deciso di seguire la stessa strada. Non so se i primi numeri di *INFO* siano mai arrivati in Italia, quindi per capire meglio il significato di questa notizia forse è il caso di fare un passo indietro.

All'inizio si chiamava *INFO 64* ed era dedicata esclusivamente al Commodore 64. Il primo numero, datato autunno 1983, era stato addirittura composto con una 9 aghi a matrice di punti, anche se poi era stata stampata su carta patinata. Solo la copertina era a colori, e rappresentava un collage di schermate del C-64. Il contenuto editoriale era rappresentato principalmente da un database che elencava tutti i prodotti disponibili a quel tempo per il C-64.

Quando arrivò sul mercato l'Amiga, la rivista estese il suo interesse anche al nuovo computer e la testata divenne semplicemente *INFO*. Più o meno in quel periodo, la pubblicazione divenne bimestrale e la redazione si trasferì in una sede più spaziosa: l'ufficio venne trasferito dalla camera da letto dell'editore a una piccola cucina riadattata. Da allora, *INFO* ha dato il via a un'azienda con un ufficio vero e proprio e ha mantenuto una periodicità bimestrale abbastanza regolare fino a oggi, per la precisione fino al numero 30 (gennaio/febbraio 1990).

Quando il numero di marzo/aprile non arrivò in edicola, cominciarono a circolare voci che parlavano di chiusura. Vennero a galla storie circa un'imminente fusione tra *INFO* e *Amigo Times* (una rivista canadese sull'Amiga, elegante e ben impaginata che, come *INFO*, viene creata interamente tramite l'Amiga). Poi si seppe che le voci circa la fusione erano fondate, ma la fusione non si fece lo stesso.

Dopo un silenzio di quattro mesi, sia *INFO* che *Amigo Times* hanno ricominciato a uscire, e sempre come pubblicazioni indipendenti. A quanto pare, *INFO* ha ottenuto i finanziamenti necessari per pubblicare 11 numeri all'anno esclusivamente dedicati all'Amiga. La rivista includerà anche una sezione tecnica che sarà curata dall'ex responsabile di *Transactor* (tra l'altro, anche *Transactor* non si vede in edicola da tre mesi). Il primo numero tutto-Amiga di *INFO* è uscito in maggio.

Amigo Times, invece, ha colto tutti di sorpresa intervenendo al World of Amiga e presentando un corposo fascicolo di 112 pagine che portava il numero 1.10. Personalmente sono felicissimo di vedere che queste due pubblicazioni sono ancora in circolazione.

Un altro sviluppo a sorpresa di questi ultimi mesi è stata la vendita delle pubblicazioni *Compute!* a Bob Guccione, editore di *Omni* e di *Penthouse*. Immediatamente hanno cominciato a circolare battute sul fatto che ora *Compute!'s Amiga Resource* e *Compute!'s Gazette* avrebbero inserito paginoni centrali con computer Commodore senza "rivestimenti esterni". Qualcuno ha ipotizzato anche l'istituzione del PET Commodore del mese.

Sfortunatamente la realtà non è così rosea. Al momento in cui sto scrivendo, la cosa più probabile è che sotto la nuova gestione *Compute!'s Gazette* chiuda i battenti, e il futuro di *Compute!'s Amiga Resource* è ancora in dubbio. Il fatto è che senza *Compute!'s Gazette*, *RUN* rimarrà l'unica rivista del nordamerica dedicata ai computer a 8 bit della Commodore. Oltretutto la salute di *RUN* è decisamente peggiore di un anno fa: l'ultimo numero aveva meno di 50 pagine ed era stampato su carta a basso costo, non patinata.

Sul fronte delle buone notizie per C-64/128, c'è l'uscita del sistema Hard Drive della **Creative Micro Designs**, che sembra molto promettente. L'Hard Drive della CMD è in grado di emulare i disk drive 1541, 1571 e 1581. I modi di emulazione sono in realtà partizioni che simulano i rispettivi disk drive. Il sistema operativo dà inoltre supporto a partizioni "indigene" che possono avere qualunque grandezza. L'unità si col-

lega alla porta seriale del computer, quella utilizzata dai normali disk drive. La velocità non dovrebbe essere un problema, dal momento che è garantito anche il supporto al Jiffy DOS, il sistema creato dalla CMD per migliorare i tempi di gestione dei drive. Per il C-128 è utilizzabile anche il modo *high speed burst*. L'Hard Drive funziona anche con *GEOS* e con il CP/M del 128, e nel complesso sembra un prodotto senz'altro molto utile. C'è un solo problema: in assenza di un attivo mercato per gli 8 bit e senza riviste specializzate ad ampia diffusione, come farà la CMD a pubblicizzare e vendere il suo prodotto?

Mostre e manifestazioni

Ultimamente mi sembra di non aver fatto altro che partecipare a mostre e presentazioni di computer. La serie è cominciata domenica scorsa con il quindicesimo Computer Festival di Trenton, che quest'anno si è tenuto nel College della Mercer County Community, anziché nel campus del College statale di Trenton com'era tradizione. Quel festival è, tra l'altro, il più vecchio e più grande "mercatino delle pulci" del Paese, per quanto riguarda i computer. Cerco sempre di esserci, perché non si può mai sapere che cosa si troverà e non c'è anno che non si concluda qualche affare. Quest'anno avrei potuto comprare un 8032 della CBM per 50 dollari e un SuperPET, completo di doppio disk drive e di stampante Commodore, per 125 dollari. Ho avuto la terribile tentazione di prenderli tutti e due, ma purtroppo non ho più un briciolo di posto per i computer che hanno superato l'età della pensione.

Ho preso invece un disk drive da 3,5", 720K, per 22 dollari, e adesso sto tentando d'interfacciarlo con l'Amiga. Mi sono portato a casa anche un paio di megabyte di RAM (a 54 dollari per megabyte) per la mia nuova scheda di memoria da 8 mega della Microbotics.

Martedì mattina (il 24 aprile) sono andato al Palladium per la festa di presentazione dell'Amiga 3000. E infine, venerdì, ho partecipato al World of Amiga a New York. È stata una settimana febbrile.

Comunque ne è valsa la pena. Queste mostre hanno sempre almeno una cosa che le rende significative: i nuovi prodotti. E il World of Amiga non ha fatto eccezione.

Il mondo dell'Amiga

Sembra che il Video Toaster della **NewTek** sia finalmente disponibile. La data ufficiale d'uscita è stata fissata per giugno, sempre che non ci siano problemi con l'approvazione della Commissione Federale Comunicazioni. La NewTek distribuiva *brochure* pubblicitarie a colori che contenevano dettagliate informazioni sul prodotto. Per 1595 dollari, il Toaster trasforma l'Amiga in una stazione video di qualità professionale: contiene quattro chip custom VLSI, accetta quattro input video e produce due output separati (principale e "preview"). Il sistema offre frame buffer da 24 bit e un generatore di caratteri con tempi di creazione di 35 nanosecondi, 16,8 milioni di colori, 25 fonti-carattere standard e la possibilità per l'utente di definire fonti personalizzate. Alcuni effetti per modificare le fonti includono la possibilità d'inserire vari tipi di ombre, la contornatura e la trasparenza.

Il Video Toaster messo "a nudo" al World of Amiga

Il Toaster offre effetti digitali in tempo reale come flip, rotazioni, ribaltamenti, blind, squeeze, zoom, mosaici, che si possono mescolare generando centinaia di effetti diversi. Il sistema può gestire, se la memoria è sufficiente, fino a mille fotogrammi. Il suo frame grabber interno può catturare fino a otto schermate consecutive (ciascuna delle quali dura un sessantesimo di secondo) e salvarle su disco. A tutte le funzioni del Toaster si accede tramite mouse, grazie a un'interfaccia zeppa di icone.

Oltre a una dimostrazione del Toaster "dal vivo", la NewTek mostrava inoltre una videocassetta promozionale che aveva come protagonisti Penn e Teller. Con il loro stile inconfondibile, i due comici spiegavano che per 1595 dollari il Video Toaster fornisce un controvalore di 60 mila dollari in apparecchiature video professionali, e che quindi per il fortunato acquirente ne deriva un profitto netto di 58405 dollari. Penso che mi convenga comprare il Video Toaster *prima* di presentare quest'argomentazione al mio consulente finanziario. Penn e Teller concludevano lo sketch proponendo la loro personale interpretazione, realizzata con il Video Toaster, di un mediocre video rock.

Allo stand della **Pulsar**, in fondo alla sala, c'era un prodotto che potrebbe diventare la novità più esplosiva dell'anno per l'Amiga 500: la scheda PC Power. "Pulsar" è un nuovo marchio lanciato da Eric Miller, distributore di prodotti per l'Amiga conosciuto anche col nome di Dr. Oxide. Molti di voi ricorderanno che Dr. Oxide era il proprietario della Comp-U-Save, una ditta che partecipava regolarmente e in tutto il mondo alle mostre Commodore.

Ora il buon dottore possiede un'azienda che diventerà presto il maggior distributore di computer Amiga, e ha abbandonato il camice verde da ospedale in favore di un elegante completo da perfetto uomo d'affari.

In ogni caso, la scheda PC Power è un clone XT fabbricato sulla base di un microprocessore NEC V30, a 8 MHz, che s'inserisce nello slot d'espansione di memoria dell'Amiga 500. Per 550 dollari, la scheda mette a disposizione il clone di un XT Turbo con grafica Hercules CGA e monocromatica, 704K di RAM che includono 64K di EMS, un BIOS Phoenix e un orologio-calendario. Il software sistema comprende l'MS-DOS 4.01, lo Shell, il GWBasic, e inoltre un database, un foglio elettronico e un word processor. L'hardware è in grado di controllare la porta seriale, quella parallela e quella mouse dell'A500. Come memoria di massa si possono utilizzare tanto i disk drive da 3,5" dell'A500 (interno ed esterni), quanto un disk drive esterno

da 5,25". Sul lato Amiga si guadagnano 512K supplementari di RAM e un RAM disk da 512K. Viene fornito inoltre il software *Cross Dos*, grazie al quale l'AmigaDOS può accedere direttamente a dischi formattati in formato MS-DOS.

La Pulsar esponeva anche un piccolo scanner a colori ($795) da usarsi con una mano, dotato di una risoluzione di 90 punti per pollice, 4096 colori e una larghezza di scansione pari a 64 mm. La lunghezza è limitata dalla disponibilità di memoria; il software permette comunque di fondere insieme strisce ottenute con scansioni successive. C'era anche uno scanner "manuale" in bianco e nero ($479) con risoluzioni di 200/300/400 punti per pollice e una scala di 16 grigi. A quanto mi risulta, inoltre, si tratta del primo scanner dotato di software OCR (Optical Character Recognition, riconoscimento ottico dei caratteri).

La Pulsar ha annunciato anche di aver acquisito i diritti di distribuzione della cartuccia Nordic Power First Amiga Action. Questa cartuccia si collega alla porta d'espansione dell'A500/1000 e prende il controllo del computer (salvando la configurazione di memoria sul disco), permette poi di eseguire una scansione della memoria alla ricerca d'immagini e suoni campionati, e infine di esaminare e modificare la memoria tramite il suo monitor interno in linguaggio macchina. Dal momento che tutti questi prodotti Pulsar arrivano dall'Europa, è possibile che ne abbiate già sentito parlare.

Continuando con gli scanner manuali, la **Migraph** ha presentato un esemplare operativo del suo sistema Hand Scanner e *Touch-Up*, che consente scansioni di quattro pollici di larghezza a 100/200/300/400 punti per pollice, tre schemi di dithering e la scansione dei tratti con un contrasto regolabile. Il software *Touch-Up* è in realtà un completo pacchetto di disegno dotato anche della capacità di convertire i file. È in grado d'importare e di esportare file IFF, IMG, PCX, TIF, GIF, e MacPaint. Le dimensioni delle immagini sono limitate soltanto dalla quantità di memoria disponibile. La Migraph dimostrava il sistema facendolo funzionare con la versione 0.91 del software. L'uscita ufficiale del sistema, con il software versione 1.0, nel momento in cui state leggendo questo articolo dovrebbe già essere avvenuta. L'interfaccia utente è molto simile a quella delle versioni disponibili per i computer MS-DOS e Macintosh.

Insieme con un'ennesima applicazione relativa alla scansione, la **Michtron** dava dimostrazione di un prototipo del Fast FAX ($699,95), un sistema di invio/ricezione di facsimili. Il Fast FAX sarà un sistema completo, software e hardware, che doterà l'Amiga di tutte le caratteristiche standard di un Fax di gruppo III. L'hardware è basato su un 68000 (funzionante a 8 MHz) con 32K di RAM e 64K di ROM. L'uscita è prevista per i primi di luglio. Della Michtron abbiamo anche visto il nuovo campionatore Master Sound, un pacchetto hardware e software.

Eric Miller "Dr. Oxide" mostra PC Power

Benché fosse citata nel catalogo della mostra, la **Beta Unlimited** di Brooklyn non è riuscita ad assicurare la sua presenza. Comunque ho ricevuto un comunicato stampa, appena prima del World of Amiga, nel quale si descriveva il sistema AudioLink, un processore audio stereo lineare a 16 bit, con 16 voci, frequenza di campionamento variabile fino a un massimo di 50 mila campioni al secondo (o centomila nel modo oversampling in mono) e una porta MIDI installata sulla scheda. Questo sistema di qualità compact-disc, che sarà disponibile per tutti i modelli dell'Amiga, sarà dotato anche di software per l'edit dei suoni. È stato progettato per garantire campioni audio di alta qualità, adatti per l'uso con una strumentazione MIDI. La Beta si augura di riuscire a portarne un prototipo funzionante all'AmiEXPO di Chicago, il 29 giugno.

Dato che stiamo parlando di campionatori, accenniamo anche a Perfect Sound 3.0 della **Sunrize Industries**, uscita circa due mesi fa. Il pacchetto è stato rivoluzionato sia come software sia come hardware. Il nuovo hardware ha due ingressi line (stereo), un ingresso microfono (mono) e la possibilità di regolare via software i livelli d'ingresso.

Al World of Amiga era presente anche la **Digital Creations**, anche se non appariva nel catalogo. Aveva portato il prototipo del nuovo sistema DCTV (Digital Composite Television), in grado di realizzare una digitalizzazione a scansione lenta di un segnale videocomposito, a colori, in circa dieci secondi. Il DCTV permette inoltre all'Amiga di visualizzare immagini a colori da 24 bit, ed è dotato di software d'elaborazione per digitalizzare le immagini e convertire immagini IFF per la visualizzazione. Il DCTV si collega sia alla porta video sia a quella parallela dell'Amiga. Si prevede che sarà disponibile per l'estate; il prezzo non dovrebbe superare i 500 dollari.

Ed eccoci al grande evento software della mostra: il debutto di *Showmaker* ($395) della **Gold Disk**. Si tratta di un pacchetto di edit multimediale che combina grafica, animazione e sonoro dell'Amiga con immagini esterne (normali o genlock) e musica prodotta via MIDI, riversando poi il prodotto finito su una videocassetta; i risultati si possono ammirare anche sul monitor dell'Amiga. Il software è dotato di un'interfaccia grafica molto intuitiva, ed è in grado di operare altrettanto bene con apparecchiature audio/video professionali o amatoriali. È possibile realizzare anche show molto lunghi, dal momento che *Showmaker* è in grado di caricare la sezione successiva mentre rappresenta quella precedente.

Per annunciare *Showmaker*, la Gold Disk ha organizzato una presentazione di un'ora che comprendeva anche un cocktail. Tra gli oratori era presente anche Kailash Ambwani, principale responsabile della Gold Disk, che ha profetizzato un grande successo negli anni Novanta per la coppia *Showmaker*/Amiga, affermando che sempre più società affidano i loro messaggi pubblicitari alle videocassette.

Hanno fatto la loro apparizione anche Guy Wright, ex direttore di

AmigaWorld e autore del volume *Amiga Desktop Video Guide* pubblicato dalla Abacus, e Lloyd Mahaffey vice-presidente per il marketing della Commodore americana. Mahaffey ha affermato che *Showmaker* della Gold Disk e il nuovo *AmigaVision* della Commodore sono prodotti complementari.

Per quanto riguarda il word processing, durante la mostra la **New Horizons** ha messo in vendita le prime copie di *ProWrite 3.0* ($175). Acquistando il programma in quell'occasione si riceveva in omaggio anche una maglietta della New Horizons (che diventerà di sicuro un pezzo da collezione). *ProWrite 3.0* aggiunge la possibilità di creare colonne multiple, un vocabolario di 300 mila parole, le macro, una porta *AREXX* e un voluminoso manuale in un involucro ad anelli nel quale si possono ammirare le sue caratteristiche di word processor WYSIWYG con grafica a colori. Ora *ProWrite 3.0* viene garantito come completamente libero da errori.

La **SoftLogik** ha fatto molti annunci al World of Amiga. *PageStream 2.0*, atteso entro giugno, sarà pienamente integrato con le fonti outline Agfa/Compugraphic così come con le fonti outline Adobe tipo I in formato IBM. Queste fonti-carattere saranno utilizzabili tanto per la visualizzazione quanto per la stampa a matrice di punti, laser o Postscript. Per gli acquirenti registrati, il prezzo dell'aggiornamento sarà di 75 dollari.

La SoftLogik ha anche appena avuto la licenza di utilizzare la raccolta di fonti-carattere Postscript e di clip art dell'Image Club Graphics. Questa raccolta è disponibile da anni per gli utenti del Macintosh, e ora farà finalmente la sua comparsa per gli utenti Amiga. La libreria di clip art contiene più di 2 mila file EPS (Encapsulated Postscript) e la raccolta di fonti ne contiene oltre 600 (Adobe tipo I compatibili).

La SoftLogik ha annunciato anche *Business Forms*, una raccolta di 40 moduli con informazioni di compilazione, in categorie come la contabilità, la fatturazione, le circolari interne e gli ordini d'acquisto.

Last but not least, la SoftLogik ha annunciato di aver dato il via a un programma di supporto per i club e per i rivenditori. I club riceveranno due copie gratuite di *PageStream*, una per sorteggiarla tra i soci e una per recensirla e utilizzarla nella preparazione dei bollettini. Il programma-rivenditori prevede un demo di *PageStream*, una copia per il negozio, uno slide-show e l'accesso gratuito alla sezione Rivenditori del BBS della SoftLogik.

C'è un nuovo nato nel settore del desktop publishing per l'Amiga, ed è la **Saxon Industries**. Alla mostra di New York ha messo in vendita la versione 1.0 di *Saxon Publisher* a 450 dollari. Questo programma ha favorevolmente impressionato gli intervenuti, mettendo in mostra una serie di capacità davvero impressionanti, alcune delle quali del tutto inedite, come i box non rettangolari per i testi e la capacità d'importare e di stampare immagini a 24 bit senza dover ricorrere a utility separate per la conversione.

Lloyd Mahaffey, vice-presidente marketing della CBM USA

Tra i partecipanti alla mostra era presente anche la **Precision Incorporated**, distributore per gli Stati Uniti di *Superbase* e *Superplan*. Nessun nuovo prodotto da parte loro, ma solo l'annuncio che la società sta attivamente lavorando per realizzare la versione Amiga di *Superbase 4*, secondo l'accordo di sviluppo stretto con la Commodore. *Superbase 4*, che attualmente funziona soltanto con *Windows* sui computer MS-DOS, dà supporto a reti locali basate su hardware Ethernet o ArcNet. La versione per l'Amiga dovrebbe essere pronta per il prossimo autunno e costerà 495 dollari.

Per i musicisti, la Dr. T's Music Software presentava ufficialmente *Tiger Cub* per l'Amiga ($99), un programma sequencer a 12 tracce, con editor grafico e notazione rapida con possibilità di stampa (si veda la presentazione di *Tiger Cub* nell'articolo "La computermusica bussa alle porte del Duemila").

Problemi da utente

E ora scambiamo qualche parola in libertà. Recentemente ho deciso di aggiungere un secondo hard disk SCSI al mio sistema, e mi sono imbattuto in un interessante problema. Sono partito con una scheda IVS Trumpcard che funzionava con un hard disk Seagate ST-157N da 48 MB come dispositivo SCSI 0. Quindi ho collegato un hard disk Quantum da 40 megabyte come dispositivo SCSI 1. Quando ho cercato di avviare l'Amiga 2000, il sistema si è limitato a bloccarsi, con il led dell'hard disk Quantum acceso. Per avviare il sistema ho dovuto disabilitare l'autoboot ed eseguirlo dal disk drive.

Ho poi scoperto che il problema consisteva nei tempi di startup dei due hard disk. Il Quantum è pronto a partire qualche secondo dopo l'accensione, mentre il Seagate non risponde a stimoli esterni ancora per venti secondi. Come risultato, quando la Trumpcard richiede il bus SCSI per il dispositivo 0 non ottiene risposta. Allora cerca il dispositivo 1, che invece risponde, e tenta di eseguire il boot dall'hard disk Quantum. Ma dal momento che lì mancano gli appropriati file per il boot, il sistema non può fare altro che bloccarsi.

Il modo più semplice per aggirare il problema, è installare i file per il boot sull'hard disk Quantum e lasciare che il sistema esegua il boot da lì. Si noti che l'Amiga esegue il boot dal Quantum, che è il dispositivo 1, soltanto al momento dell'accensione. Ma quando viene fatta una "warm start" (un reboot a computer già acceso), tramite la contemporanea pressione dei tasti Control/Amiga-destro/Amiga-sinistro, il boot viene eseguito dall'hard disk Seagate. La soluzione giusta sarebbe quella d'impostare il Quantum come dispositivo SCSI 0 e il Seagate come dispositivo SCSI 1. Ora come ora, sto lavorando con un'interfaccia SCSI Supra Wordsync e un'interfaccia SCSI ALF. La prossima volta vi farò sapere se anche con quelle si verifica lo stesso tipo di problema. ■

LA GESTIONE DELLA MEMORIA IN LINGUAGGIO C

Approfondiamo l'uso delle funzioni AllocMem, FreeMem, malloc, free, sbrk, exit, Allocate, Deallocate e realloc

di John Foust

La forma più semplice e corretta di gestione della memoria è l'allocazione dinamica, e il linguaggio C gestisce questo processo ricorrendo alle funzioni AllocMem() e FreeMem() della libreria Exec, e alle proprie funzioni interne malloc() e free(). Con AllocMem() si crea un array che potremmo chiamare *values*, modificabile tramite una variabile numerica intera chiamata *num*. AllocMem() ha due argomenti: il numero di byte di memoria e il tipo di memoria (i flag che specificano il tipo di memoria, come MEMF_PUBLIC, sono elencati nel file INCLUDE exec/memory.h). Prendiamo come esempio un programma che esegua calcoli numerici, e utilizzi la funzione AllocMem() per richiedere la memoria di cui ha bisogno.

```
int Calculate(num)
int num;
{
int success;
int *values;
ULONG val_sz;
  /* Trova la grandezza dell'array in byte */
  val_sz=(ULONG) num * sizeof(int);
  /* Quindi richiede la memoria */
  values=AllocMem(val_sz, MEMF_PUBLIC);
  if  values !=NULL) {
      success=TRUE;
    /* A questo punto si inseriscono i calcoli */
    FreeMem(values, val_sz);
  }
  else {
    success=FALSE;
  }
  return success;
}
```

Calculate() segue una procedura particolarmente accurata nell'allocazione della memoria. Se AllocMem() restituisce NULL, il programma non effettua i calcoli. Se invece trova memoria sufficiente, utilizza la variabile *values* come se avesse a che fare con un array statico, in espressioni come:

```
values[5]=34 * values[4],
```

Per questo tipo di array non è necessario dichiarare un particolare contenuto (nemmeno di strutture), ma è importante valutare correttamente le dimensioni del blocco di memoria (tramite l'operatore sizeof).

Si osservi che il fallimento di AllocMem() si riflette inevitabilmente in un fallimento del programma, per quanto insignificante sia l'ammontare di memoria. Calculate() restituisce un valore TRUE se la memoria può essere allocata, altrimenti restituisce FALSE. Una funzione che chiama Calculate() deve controllare il valore restituito per assicurarsi che nel calcolo tutto sia andato liscio. Se non è così, il programma deve avvertire l'utente che la mancanza di memoria ha causato un errore. In programmi complessi, un errore nella sotto-routine di più basso livello deve "infiltrarsi" verso l'alto fino a raggiungere una funzione di alto livello in grado di correre ai ripari, e le funzioni interessate devono cancellare la memoria da esse allocata.

La memoria allocata viene restituita al sistema tramite FreeMem(). L'esempio precedente chiama FreeMem(), indicando come argomenti un puntatore alla memoria e un puntatore alle dimensioni della memoria. Con FreeMem() il programma deve "ricordare" le dimensioni del blocco allocato.

Ogni libreria C standard contiene due funzioni, malloc() e free(), che hanno una stretta parentela con AllocMem() e FreeMem() dell'Exec. Sono funzioni particolarmente "adattabili", dal momento che permettono la compilazione di un programma su macchine molto diverse (non è difficile adattare l'esempio precedente per usarlo con malloc).

Le routine malloc() e free() sono scritte in termini di AllocMem() e FreeMem(). Nel compilatore C della Manx, una chiamata di malloc() per 32 byte si traduce in una chiamata ad AllocMem per 40 byte. Il sistema impiega la memoria supplementare per mantenere una lista concatenata che tiene il conto della memoria allocata al fine di poterla liberare automaticamente al momento opportuno. Questo semplice schema di memoria è illustrato nel file misc/malloc.c, contenuto nel codice sorgente della libreria Manx.

Molti compilatori C non si limitano a chiamare AllocMem() in corrispondenza di ogni chiamata a malloc(), ma mantengono una lista di memoria personale. Le routine che svolgono questo compito imitano in genere le funzioni Unix di basso livello per la gestione della memoria, come sbrk(). Per avere questo tipo di gestione nel sistema Manx si deve eseguire il link del file heapmem.o, mentre il Lattice lo offre come default. La prima chiamata a malloc() per ottenere memoria dal sistema operativo ha l'effetto di chiamare la funzione sbrk(), e quest'ultima impiega AllocMem() per richiedere un ampio blocco di memoria (con il Manx la quantità

di default è 40K). Ogni successiva chiamata a malloc() sfrutta una parte di questo blocco iniziale.

Per le librerie Manx, si consulti il file sysio/heapmem.c per ulteriori chiarimenti sull'interazione tra malloc(), AllocMem() e sbrk(). La grandezza del blocco è regolabile tramite la variabile a_Heapsize, che può essere dichiarata come "extern unsigned long _Heapsize;" se si desidera effettuare il reset del suo valore nei propri codici. Si noti che è necessario impostarla *prima* di chiamare malloc() o le funzioni di I/O standard. A questo scopo, il Lattice impiega _MNEED e MSTEP.

Nel Manx viene allocato soltanto un grosso blocco iniziale; nel Lattice, invece, le routine gestiscono una lista di blocchi. Quando un blocco di 40K è completamente sfruttato, malloc() chiama sbrk() per allocarne un altro, minimizzando le chiamate ad AllocMem(). Le chiamate a free() "etichettano" porzioni di questi blocchi per riutilizzarle; FreeMem() non viene mai chiamata. A differenza di FreeMem(), free() ha bisogno soltanto di un puntatore alla memoria, e non richiede le dimensioni del blocco allocato. Queste dimensioni, infatti, sono contenute in una struttura posta immediatamente prima dell'effettivo blocco di memoria allocata.

Le funzioni delle librerie C standard forniscono anche un meccanismo automatico di "pulizia". Qualunque cosa dica il codice sorgente, i programmi in linguaggio C non iniziano dalla cima della routine main() né finiscono all'ultima riga di main(). Vi è sempre una certa parte di codici di libreria che vanno in esecuzione prima e dopo l'esecuzione di main(): la main() del programmatore viene in realtà chiamata da una funzione di libreria che si chiama _main(); e quando la main() del programmatore conclude la sua esecuzione, il controllo torna a _main(). All'interno di _main(), tutta la memoria allocata tramite malloc() viene automaticamente restituita da una funzione di libreria standard chiamata exit(), come salvaguardia nei confronti di una programmazione poco accurata. Nel corso dell'operazione di pulizia eseguita da exit(), le routine di gestione della memoria in linguaggio C mandano in esecuzione FreeMem() su tutti i grandi blocchi di memoria, ignorando le informazioni ivi contenute.

Questa ripulitura automatica, sia chiaro, non vi libera dalla responsabilità di rilasciare la memoria allocata. Quando al termine di un programma rimane della memoria, si deve sempre restituirla al sistema tramite free() oppure FreeMem(). Se però va storto qualcosa, chiamate pure exit() e tutta la memoria allocata da malloc() viene restituita al sistema.

Anche se malloc() e free() rendono più semplice qualche aspetto della programmazione, non possono essere utilizzate per qualunque compito dell'AmigaDOS. Per esempio, le immagini grafiche devono essere immagazzinate nella memoria chip, e malloc() non fa differenza tra memoria chip e fast; è quindi necessaria AllocMem(). Fate attenzione: mescolare chiamate a malloc() e ad AllocMem() nello stesso programma presenta qualche rischio. La prima chiamata a malloc(), infatti, alloca una grande quantità di memoria che non viene restituita al sistema fino al momento dell'uscita dal programma, anche nel caso che il programma ne impieghi soltanto una piccola parte.

Comunque anche AllocMem() crea problemi. Infatti, una serie di piccole allocazioni di memoria fatte da diversi programmi in tempi ravvicinati dà origine a un fenomeno di frammentazione. In altre parole, la memoria libera del sistema viene frazionata in molti piccoli frammenti non contigui. Le allocazioni di memoria (da parte di programmi diversi) si intrecciano l'una con l'altra, e quando un programma conclude la sua esecuzione lascia dietro di sé delle "falle" di memoria libera. In questa situazione, le richieste di grandi blocchi continui sono destinate a fallire, perché i blocchi disponibili sono troppo piccoli. Per evitare la frammentazione, l'Exec mette a disposizione le routine Allocate() e Deallocate(), che rendono più semplice scrivere codici dotati di un livello intermedio di gestione della memoria. Allo stesso scopo esiste anche una funzione standard del C, realloc(), che cambia la grandezza di un blocco di memoria e copia nel nuovo blocco i dati contenuti nel vecchio. Si devono fornire a realloc() due argomenti: un puntatore a un blocco di memoria precedentemente allocato tramite malloc() e la nuova grandezza del blocco di memoria. La funzione restituisce un puntatore al nuovo blocco. Assicuratevi di utilizzare questo puntatore per riscrivere le assegnazioni delle variabili che puntavano a quel blocco di memoria. È senz'altro possibile scrivere una propria versione di realloc() utilizzando AllocMem() e FreeMem(); dal momento però che AllocMem() ha bisogno di un flag che indichi il tipo di memoria, e FreeMem ha bisogno di conoscere la grandezza del blocco di memoria da rilasciare, questa funzione personalizzata sarà sicuramente più complicata di realloc(). Ecco un esempio:

```
/* reallocmem.c */
#include "exec/types.h"
#include "exec/memory.h"
#define MIN(a,b) ((a)<(b)?(a):(b))
void *ReAllocMem(ptr, newsize, odsize, type)
void *ptr;
ULONG newsize, oldsize, Type;
{
void *new_ptr;
ULONG copysize;
  /* Alloca un blocco di grandezza diversa */
  new_ptr=AllocMem(newsize, type);
  if (new_ptr !=NULL){
  /* Copia i vecchi dati nel nuovo blocco */
  copysize=MIN(newsize, oldsize)
  CopyMem(ptr, new_ptr, copysize);
  /* Rilascia il vecchio blocco */
  FreeMem(ptr, oldsize);
}
return new_ptr;
}
```

Se non si può effettuare la riallocazione, allora il vecchio blocco rimane com'era, e ReAllocMem() restituisce NULL. Se invece la riallocazione ha successo, il vecchio blocco viene automaticamente rilasciato. Si noti che l'effetto del flag MEMF_CLEAR non viene preso in considerazione da ReAllocMem(), perché ogni vecchio dato viene copiato nel nuovo blocco. Se il nuovo blocco è più grande, MEMF_CLEAR garantisce che la parte in eccedenza venga impostata a zero. Tramite ReAllocMem(), cambiare la grandezza degli array dinamici durante l'esecuzione di un programma diventa un'operazione piuttosto semplice. Non vi resta proprio nessuna scusante per una scadente gestione della memoria. ■

COMINCIAMO DA ZERO: PARTE SECONDA

Facciamo conoscenza con i cassetti, gli strumenti e i progetti

di Mark L. Van Name e Bill Catchings

Il *Workbench* tratta ogni disco come se fosse un archivio che può contenere tre tipi di oggetti: progetti (project), strumenti (tool) e cassetti (drawer).

Un *progetto* è in sostanza l'equivalente elettronico del "fascicolo" di un normale archivio. In genere si tratta di un file creato con un foglio elettronico, con un database, con un word processor o con un programma di disegno... ovvero qualunque cosa venga immagazzinata in memoria e possa essere manipolata dal computer.

Per "manipolare" un progetto è necessario uno *strumento*, che in genere è un breve programma. Può essere un programma contenuto nel disco del *Workbench*, come nel caso degli strumenti *Notepad* e *Calculator*, oppure può essere un programma acquistato separatamente e creato da programmatori esterni alla Commodore.

Un *cassetto* è semplicemente un luogo di "immagazzinamento" dotato di un nome. Può contenere strumenti, progetti e persino altri cassetti, permettendo all'utente di organizzare le informazioni gerarchicamente. Si può avere per esempio un cassetto per le dichiarazioni dei redditi, e al suo interno una serie di cassetti per ogni anno; poi, all'interno di ogni cassetto "annuale" si colloca una pagina di foglio elettronico contenente la corrispondente dichiarazione dei redditi.

Tutti e quattro questi oggetti fondamentali (dischi, cassetti, strumenti e progetti) possiedono icone che l'utente vede sullo schermo e che può modificare (perlomeno in ambiente *Workbench*). Come forse già saprete, l'Amiga è come un animale a due teste: oltre al *Workbench*, che è basato sulle icone, esiste un'interfaccia-utente costituita di una linea comandi (che viene spesso chiamata semplicemente CLI oppure, nella versione 1.3 dell'AmigaDOS, "Shell"). Il CLI utilizza gli stessi oggetti fondamentali del *Workbench* ma a volte dà loro nomi differenti. I dischi sono sempre dischi, ma i cassetti sono ribattezzati "directory". Inoltre, il CLI chiama "file" tanto i progetti quanto gli strumenti, facendo tra gli uni e gli altri una sola differenza: alcuni file (gli strumenti) si possono mandare in esecuzione come programmi e altri no (i progetti).

In ambiente CLI, file e directory possiedono ancora icone, ma queste icone sono semplicemente file speciali con nomi particolari (il "nome iconico" di un file è lo stesso nome del file o della directory, a cui viene però aggiunto il suffisso .info; per esempio, il nome iconico di *Notepad* è Notepad.info). Il CLI comunque permette anche l'esistenza di file e directory privi di icone. Vi sono molti file di questo tipo nel disco del *Workbench*, e nessuno di loro compare sullo schermo Workbench. I soli file visibili sullo schermo Workbench sono quelli dotati di icona.

Torniamo per un attimo al *Workbench*. Il *Workbench* mette a disposizione parecchi modi per avere informazioni sugli oggetti-disco: il modo più semplice è "aprire" l'oggetto stesso. Per farlo vi sono due sistemi: *selezionarlo* (ovvero collocare il puntatore sull'icona relativa e premere una volta il pulsante sinistro del mouse) e quindi scegliere Open dal menu Workbench, oppure *effettuare un doppio click* (ovvero premere il pulsante del mouse due volte in rapida successione, sempre dopo averlo collocato sull'icona).

Quando si apre un disco oppure un cassetto, compaiono i cassetti o i file dotati di icona contenuti al suo interno. Se per esempio si apre il disco *Workbench* compaiono parecchi cassetti, uno dei quali si chiama Utilities. File come *Notepad* o *Calculator* si trovano appunto nel cassetto Utilities.

Aprire invece uno strumento (tool) o un progetto produce un risultato diverso: quando si apre uno strumento, il *Workbench* lo manda in esecuzione, ma senza indirizzarlo a nessun particolare progetto. A questo punto l'utente dovrebbe caricare il progetto sul quale intende lavorare. Quando si apre un progetto, il *Workbench* sa che è necessario un particolare strumento per esaminare quello che c'è al suo interno. Lo strumento in questione è noto come "tool di default del progetto"; in genere si tratta del tool utilizzato per creare il progetto. Il *Workbench* manda automaticamente in esecuzione il tool di default sul progetto: in tal modo l'utente può esaminarne il contenuto e decidere se intende continuare a lavorarvi.

Il fatto che il *Workbench* "conosca" ogni tool di default dei vari progetti è soltanto una delle varie informazioni che il sistema memorizza su ogni oggetto. Tutte le informazioni di questo tipo si possono visualizzare "selezionando" (vedere sopra) il particolare oggetto e scegliendo l'opzione Info dal menu Workbench: quest'operazione fa apparire una finestra informativa sull'oggetto.

Tra i dati visualizzati sulla finestra vi sono: il **nome** (name) dell'oggetto; il **tipo** (type), che può essere "disco" (disk), "cassetto" (drawer), "progetto" (project) o "strumento" (tool); una riga chiamata **tipi di strumenti** (tool types), che spiega semplicemente quali altri strumenti possono essere utilizzati per operare sull'oggetto; un'area chiamata **stato** (status), il cui contenuto varia a seconda del tipo di oggetto (la stessa cosa vale per tutti gli altri dati contenuti nella finestra).

Nei dischi l'area "stato" indica se il disco è a lettura/scrittura oppure a sola lettura, ovvero se è stato protetto in scrittura spostando in posizione sollevata l'apposita linguetta di protezione. In una seconda area

sono presenti alcune informazioni sulla natura del disco, tra cui **byte per blocco** (byte per block), ovvero un numero che per i dischi da 3,5" vale 488; **numero di blocchi** (number of blocks), ovvero la quantità totale di blocchi presenti sul disco; **numero utilizzato** (number used), ovvero la quantità di blocchi già occupati da informazioni; **numero libero** (number free), ovvero la quantità di blocchi disponibili. Una terza area, chiamata **strumento di default** (default tool), indica che lo strumento di default per i dischi è il programma Diskcopy (in effetti si legge: "SYS: System/Diskcopy", che è il nome completo attribuitogli dal CLI).

I cassetti non possiedono uno strumento di default, ma al suo posto hanno una linea-commento (comment) dove l'utente può inserire un breve promemoria sul motivo per cui ha creato il cassetto o sulle informazioni che vi sono state immagazzinate. I cassetti hanno un'opzione "stato" analoga a quella degli strumenti e dei progetti, ma diversa da quella dei dischi. L'area "stato" di cassetti, strumenti e progetti mostra quattro possibilità: ARCHIVIATO (archived), ABILITATO IN LETTURA (readable), ABILITATO IN SCRITTURA (writeable), CANCELLABILE (deleteable); le ultime tre voci dicono quali operazioni si possono effettuare sull'oggetto, mentre la prima informa se l'oggetto è stato salvato nel corso dell'ultima sessione di lavoro. Ogni possibilità che non risulta vera viene evidenziata e preceduta da un NOT.

Le informazioni contenute nella finestra informativa possono anche essere modificate. Per esempio, si può proteggere un oggetto dalla cancellazione portando il puntatore su DELETEABLE e premendo il pulsante del mouse: la scritta diventa NOT DELETEABLE. Se poi si preme il pulsante del mouse su SAVE (salva), in basso a sinistra, ogni successiva chiamata dell'opzione Info rifletterà il cambiamento di stato dell'oggetto. Per uscire da questo schermo senza effettuare nessun cambiamento ci sono due alternative: si può premere il pulsante del mouse su QUIT (abbandona), in basso a destra, oppure sul gadget di chiusura situato nell'angolo superiore sinistro della finestra.

Come i cassetti, anche gli strumenti e i progetti hanno nelle loro finestre informative un'area (chiamata COMMENT) destinata a ricevere un promemoria. In più hanno un'area SIZE (dimensioni) che indica lo spazio da essi occupato sul disco, sia in byte sia in numero di blocchi. I progetti, come i dischi, hanno un DEFAULT TOOL e un'area TOOL TYPES; possiedono inoltre un'altra linea, chiamata STACK, che permette all'utente di stabilire quanta memoria di stack utilizzare nel corso del lavoro con quel particolare progetto. Quando avrete un po' più di esperienza, vi capiterà spesso di modificare il valore dello stack.

Prima o poi si manifesterà l'esigenza di cambiare il nome di un oggetto. Quest'operazione non può essere svolta dalla finestra informativa (non si possono infatti digitare caratteri nell'area NAME), ma si svolge direttamente dal *Workbench*. Per modificare il nome di cassetti, progetti o strumenti, s'impiega la stessa procedura illustrata nel numero scorso per cambiare il nome di un disco. Prima di tutto si seleziona l'oggetto e poi si seleziona Rename dal menu Workbench, facendo apparire sullo schermo un gadget stringa che contiene il nome dell'oggetto, si modifica il nome e si preme Return. Questo è tutto.

Nel prossimo articolo ci occuperemo di qualche altra operazione fondamentale che il *Workbench* permette di effettuare su dischi, cassetti, strumenti e progetti. ■

CORSO DI PROGRAMMAZIONE IN C PER L'AMIGA: OTTAVA PUNTATA

LA LIBRERIA DOS/ARP AL MICROSCOPIO: PRIMA PARTE

Un attento esame al microscopio delle prime funzioni contenute nella libreria DOS/ARP, quelle direttamente riferite a file e directory

di Eugene P. Mortimore

Nel precedente articolo di questa serie ("Le procedure d'azione nei programmi Intuition") abbiamo presentato un programma d'esempio che ricorreva ad alcune funzioni della libreria DOS/ARP per gestire i file di un database. Quel programma forniva una semplice dimostrazione di come si possono gestire i file in un programma Intuition, inserendo e visualizzando un testo in una finestra.

Arrivati a questo punto, conviene abbandonare temporaneamente il regno di Intuition e indagare sulle caratteristiche intrinseche delle funzioni DOS della libreria ARP (ovvero le prime 33). I programmi che le utilizzano si possono definire "programmi di gestione di directory e file". Come sempre, il modo migliore per scoprire come si comporta una funzione è vederla all'opera in un programma d'esempio.

I file di un programma possono essere grossomodo classificati come file di I/O *primari*, file *temporanei* (file che vengono scritti automaticamente dal programma in particolari occasioni), file *di configurazione* (che permettono all'utente di personalizzare le azioni svolte dal programma) e file di I/O *secondari* (che il programma può richiedere in certe circostanze operative).

Se utilizzate per i vostri programmi il compilatore C della Manx, osserverete che questo sistema opera con file che appartengono a ciascuno dei quattro tipi elencati (alcuni li richiede in input e altri li genera in output).

Per realizzare un sistema di gestione di file e directory, è importante capire la necessità di un'appropriata interazione con l'utente. In genere questo significa, tra l'altro, che il programma deve sforzarsi d'individuare tutti gli errori che l'utente potrebbe commettere per una non perfetta comprensione del sistema o per distrazione. Ci si deve sempre basare sull'idea che l'utente non sappia nulla di programmi, dischi, disk drive e di tutti i concetti che i programmatori tendono a considerare universalmente noti. Portando questo atteggiamento all'estremo, conviene presupporre che l'utente non abbia letto con particolare attenzione il manuale del programma, o addirittura che non l'abbia letto del tutto.

Il programma deve prendere in considerazione almeno questi aspetti dell'interazione tra utente e programma: (1) quanti e quali dischi l'utente dovrà utilizzare; (2) in quali istanti dev'essere inserito un certo disco in un particolare disk drive; (3) l'esatta sequenza d'inserimento dei dischi; (4) i tipi di errore che l'utente potrebbe fare (per esempio inserire il disco sbagliato al momento sbagliato); (5) quali messaggi dovrebbero essere inviati all'utente (tramite la finestra CLI o i requester di Intuition) per aiutarlo a usare correttamente il programma; (6) come deve reagire il programma alle contromisure che potrebbe prendere l'utente in seguito alla segnalazione di un errore.

In un certo istante, per esempio, il programma può aver bisogno che un particolare disco si trovi nel drive DF1:, e l'utente può invece non averlo inserito o averlo rimosso troppo presto. Il programma deve quindi essere in grado di rilevare la mancanza e d'informare l'utente sulle contromisure da prendere.

Un'altra possibilità è che il programma cerchi di cancellare un file in un disco protetto. Ancora una volta il programma deve capire cos'è successo e quindi suggerire all'utente di spostare la linguetta di protezione del disco o d'inserire nel disk drive un altro disco non protetto.

Oppure, il file che il programma intende cancellare potrebbe essere stato protetto dalla cancellazione tramite un comando impartito a livello CLI o tramite una chiamata alla funzione SetFileProt. In un caso simile, il programma deve chiedere all'utente se conferma la cancellazione del file. Se l'utente risponde S (sì) il programma deve rimuovere la protezione chiamando la funzione SetFileProt e poi chiamare la funzione DeleteFile.

Tutti questi casi (e potremmo citarne molti altri) richiedono che il programma abbia previsto in anticipo il possibile comportamento dell'utente e i possibili errori

fisici nel sistema dei dischi (per esempio un errore nella ricerca di una specifica posizione, a causa di un'imperfezione sulla superficie del disco).

Quando il programma chiama una funzione DOS, deve per prima cosa controllare se la chiamata ha avuto successo. A questo scopo esamina il valore restituito dalla chiamata, e lo colloca in un'opportuna variabile locale. Questo valore viene chiamato "errore primario". Se l'errore primario è in realtà una condizione di non-errore, allora il programma può essere sicuro che la chiamata ha avuto successo e passa all'istruzione successiva.

Se invece il valore individuato è diverso da quello atteso, il programma deve determinare la causa dell'errore chiamando la funzione IoErr e leggendo il valore da essa restituito. Quest'ultimo valore viene chiamato "errore secondario".

Ci sono quindici tipi di errori che riguardano la gestione di file e directory. Il nostro programma d'esempio spiega come devono comportarsi i programmi per correre ai ripari.

Funzioni DOS è un programma in linguaggio C che ricorre a tutte le funzioni elencate nella Tavola 1 dell'articolo "Le funzioni DOS per la gestione di file e directory", pubblicato in questo stesso numero. Il programma è composto da un segmento principale e da una sequenza di quindici chiamate ad altrettante routine, e descrive le principali caratteristiche dei valori di chiamata-e-ritorno (ed elaborazione degli errori) delle principali funzioni DOS relative alla gestione di directory e file. È evidente che il programma è stato progettato a scopo didattico, per studiare i dettagli e i requisiti di ogni singola chiamata; le quindici chiamate, infatti, non hanno alcuna relazione tra loro.

Come ipotesi di lavoro si suppone che il programma (e tutti i file a esso relativi) si trovino in un disco inserito in DF0: o DF1:. Il programma inoltre crea e gestisce tre directory e nove file in DF1:.

Per una corretta compilazione, nella directory DF0:INCLUDE devono esserci i file INCLUDE arpbase.h, arpfunc.h e gli altri indicati all'inizio del listato. Abbiamo eseguito la compilazione con il sistema Manx 3.6A e abbiamo impiegato il comando *set* per descrivere le variabili ambientali di sistema, dicendo al compilatore dove si trovavano i file INCLUDE e le librerie linked.

Ecco un esempio d'impiego del comando set:

```
set INCLUDE=DF0:include!DF1:include
INCLUDE=DF0:lib!DF1:lib
```

Per mandare in esecuzione il programma dovete organizzare i file nel modo seguente: in DF0:LIBS la libreria ARP; in DF0:ROOT o in DF1:ROOT il programma stesso.

Le quindici routine chiamate dal programma sono nell'ordine: MakeDir, OpenFile, WriteFileText, ReadFileText, SeekEOF, AddNewFileText, SetDirComment, SetFileComment, SetDirProt, SetFileProt, TryDeleteDir, TryDeleteFile, TryRenameDir, TryRenameFile e CloseFile.

Per aiutare a capire come funziona il programma nelle diverse circostanze, abbiamo creato un file di riepilogo degli errori, che registra tutti i principali errori verificatisi durante l'esecuzione. Se il programma si conclude prematuramente, si può utilizzare il proprio editor per vedere quale errore ha causato l'uscita.

In particolare, il file di riepilogo registra il verificarsi degli errori che provocano il fallimento delle chiamate di funzione. Tutte le voci del file di riepilogo provengono da chiamate alla funzione LogError, che a sua volta chiama la funzione C standard "fputs" (che colloca una stringa in un file).

Se per esempio tentate di mandare in esecuzione il programma senza la presenza della libreria ARP nella directory DF0:LIBS (o DF1:LIBS), il programma usa fputs per registrare una voce nel file di riepilogo ("Could not open ARP library"; non posso aprire la libreria ARP) poi cede il controllo alla finestra CLI di partenza. La stessa cosa accadrebbe, con un messaggio diverso, se il programma (dopo aver trovato la libreria ARP) non riuscisse a chiamare MakeDir, perché nel frattempo avete tolto il disco da DF1:, oppure perché il disco presente in DF1: non è quello giusto.

Un esame del listato

All'inizio del listato ci sono tutti i riferimenti ai suoi file INCLUDE. Il programma ne richiede sei:
- stdio.h serve per le chiamate alle funzioni C standard di I/O: printf, fputs, fopen, fclose e strlen.
- exec/types.h è necessario per definire i tipi di dati specifici per l'Amiga che vengono impiegati nel programma, come per esempio BOOL (valore booleano).
- dos.h e dosextens.h contengono tutte le informazioni specifiche dell'Amiga (costanti, definizioni di strutture e valori degli errori) richieste per interfacciarsi con le funzioni DOS/ARP.
- arpbase.h contiene tutte le informazioni (costanti, definizioni di strutture e valori degli errori) necessarie per dare supporto alle funzioni della libreria ARP.
- arpfunc.h contiene altre informazioni vitali sulle funzioni della libreria ARP.

Vi è poi un elenco di definizioni #defines scelte per migliorare la leggibilità del programma. Sono state create nuove abbreviazioni per i nomi degli errori generati dal programma. Tutti i nomi originali degli errori (meno uno) sono definiti nel file INCLUDE dos.h. Altre abbreviazioni sono state definite semplicemente per rendere più sintetiche le righe del programma.

Subito dopo gli #include e i #defines vengono i dati globali, che sono variabili il cui contenuto è sempre disponibile per ogni funzione del programma: tutte le quindici funzioni "vedono" in ogni momento il loro valore. Per la maggior parte, queste variabili (per esempio dirName[i], fileName[j]) sono semplici stringhe di testo che non cambiano mai, mentre altre (come readFileText[][]) si comportano come RAM buffer per scrivere (e forse in futuro per leggere) stringhe di testo variabili.

I dati globali sono formati principalmente da array di nomi di variabili locali, e la maggior parte delle voci degli array sono definite come normali stringhe a terminazione nulla che fanno uso della solita notazione tra virgolette del C. Esaminando queste stringhe, si vede subito che il programma gestisce tre directory e nove file in DF1:.

Ecco quali sono le variabili globali definite:
- una variabile array (dirName) che definisce i nomi delle tre directory da creare e quindi gestire su DF1:;
- una variabile array (newDirName) che definisce i nuovi nomi per le tre directory originali su DF1:;

– una variabile array (dirComment) che definisce un gruppo di tre stringhe di commento alle tre directory;
– un intero LONG (DirProtMode) che il programma utilizza per impostare la protezione contro la cancellazione per le tre directory;
– una variabile array (FileName) che fornisce le stringhe utilizzate dal programma per i nomi dei nove file;
– una variabile array (NewFileName) che fornisce i nuovi nomi per i nove file del programma;
– una variabile array (fileText) che contiene gli effettivi testi che il programma colloca (scrive) in ciascuno dei nove file;
– una variabile array (newFileText) che contiene gli effettivi testi che il programma aggiunge al termine dei contenuti originali dei nove file;
– una variabile array (fileComment) che definisce un gruppo di tre stringhe di commento per i nove file;
– un intero LONG (FileProtMode) che imposta una protezione contro la cancellazione dei file;
– una variabile array consistente in un RAM buffer di dati (readFileText) che il programma impiega come RAM buffer per mantenere i testi dei nove file;
– una variabile array formata da tre puntatori (dirLock) che puntano alle strutture FileLock delle nuove directory;
– una variabile array formata da tre puntatori (oldDirLock) che puntano alle strutture FileLock delle vecchie directory;
– una variabile array formata da nove puntatori (fileLock) che puntano alle strutture FileLock dei vecchi file;
– una variabile array formata da nove puntatori (fileHandle) che puntano alle strutture FileHandle dei nuovi file;
– un intero LONG (EN) che contiene i numeri degli errori restituiti dalla funzione IoErr;
– due interi SHORT ("i" e "j") che il programma impiega in molti punti come indici per i loop "for";
– un intero SHORT (OF) che il programma impiega per indicare che i nove file sono aperti;
– un valore booleano (success) che il programma controlla per determinare se una particolare chiamata si è conclusa con successo;
– una variabile (fP) che punta a una struttura FILE e viene utilizzata dal programma per inserire nel file di riepilogo le informazioni sugli errori.

La prima routine chiamata da main è **MakeDir**. Ha un comportamento molto semplice: esegue un loop "for" che crea tre directory in DF1:. Svolge il suo compito chiamando la funzione CreateDir della libreria DOS e usa come unico argomento la variabile globale dirName[i].

CreateDir restituisce un puntatore alla struttura FileLock; se il puntatore vale zero, MakeDir chiama IoErr per avere informazioni sull'errore, quindi confronta il valore restituito da IoErr con sei specifici valori d'errore. Se uno di questi è uguale al valore restituito da IoErr, MakeDir chiama LogError e il programma interrompe l'esecuzione, tornando alla finestra CLI di partenza e visualizzando le opportune informazioni circa l'errore che ha causato il problema.

Se invece la funzione riesce a creare con successo tutte e tre le directory in DF1:, visualizza tre messaggi sulla finestra CLI di partenza e restituisce il controllo al programma main.

Dopo la creazione delle tre directory, il programma main chiama la routine **OpenFile** che ha il compito di aprire nove file in DF1: (tre per ognuna delle directory appena create).

Per prima cosa OpenFile imposta a 0 la variabile OF (si ricordi che OF è la variabile globale che indica se tutti e nove i file sono stati aperti con successo: impostare OF a 0 significa che nessun file è aperto).

Subito dopo, OpenFile entra in un ciclo "for" e chiama la funzione Open della libreria DOS utilizzando l'argomento fileName[j] come puntatore al nome del file da aprire. Il secondo argomento della chiamata è MODE_NEWFILE, a indicare che il programma sta aprendo file che prima non esistevano. Open restituisce un puntatore a una struttura FileHandle; se il puntatore vale 0, significa che il file non può essere aperto. In tal caso OpenFile chiama IoErr per avere informazioni sull'errore: se il valore restituito coincide con uno dei cinque valori noti alla routine, OpenFile chiama LogError e il programma si conclude prematuramente.

Se invece tutti i file vengono aperti con successo, il programma visualizza nove messaggi nella finestra CLI di partenza e restituisce il controllo a main.

Il programma main chiama quindi **WriteFileText** per associare un testo a ciascuno dei nove file (il testo da associare è quello dichiarato in precedenza nella variabile globale fileText). WriteFileText inizia chiamando la funzione in linguaggio C "strlen" per determinare il numero di caratteri contenuto in ciascuna stringa di testo (questo valore viene utilizzato come terzo argomento nella chiamata alla funzione Write della libreria DOS). Il secondo argomento è fileText[j], che punta a una stringa di testo, e il primo argomento è fileHandle, puntatore alla struttura FileHandle relativa al file.

La funzione Write restituisce un numero nella variabile actuallength: se vale –1, il programma riconosce una condizione d'errore e chiama IoErr per saperne di più. Se il numero restituito da IoErr coincide con uno dei sei valori d'errore noti alla funzione, il programma chiama LogError per registrare l'errore nel file di riepilogo. Se invece la scrittura dei nove file si conclude con successo, il programma visualizza nove messaggi nella finestra CLI di partenza e visualizza inoltre le nove stringhe di testo inserite nei file.

Il programma main passa quindi a chiamare la routine **ReadFileText**, che legge le stringhe di testo appena inserite e le memorizza in altrettanti RAM buffer. La struttura di questa routine è molto simile a quella della precedente: come WriteFileText, anche ReadFileText controlla il valore di actuallength e se vale –1 chiama IoErr. Confronta quindi il numero restituito da IoErr con quattro valori d'errore, e nel caso che uno dei quattro coincida, chiama LogError per registrare l'errore nel file di riepilogo. Se la lettura avviene invece senza errori, il programma visualizza nove messaggi nella finestra CLI di partenza e restituisce il controllo a main.

Le altre dieci funzioni chiamate da main sono strutturate in modo molto simile a quelle già descritte: per trarre dal programma la massima utilità è sufficiente capire quali argomenti richiedono e quali valori restituiscono. Si veda anche l'articolo "Le funzioni DOS per la gestione di file e directory", dove sono trattate in dettaglio le funzioni DOS relative a file e directory.

Come si è visto, abbiamo fatto in modo che a ogni chiamata di LogError il programma tornasse alla

finestra CLI di partenza. Se però esaminate gli errori uno per uno, noterete che non tutti dovrebbero interrompere l'esecuzione del programma. A questo scopo si potrebbero scrivere due funzioni LogError, una per gli errori "fatali" e una per quelli "non-fatali" (errori di avvertimento), e fare in modo che venga chiamata l'una o l'altra a seconda del tipo d'errore.

Se per esempio si verifica un errore di scrittura in un file protetto, si può far intervenire l'utente costruendo una funzione LogError che gli chieda (tramite l'alternativa S-N) se vuole che il file sia sovrascritto oppure no.

Oppure, se si verifica un errore di scrittura su un disco protetto si può suggerire all'utente di ripetere l'operazione dopo aver spostato la linguetta di protezione. In tal caso si deve costruire una funzione LogError per chiedergli se intende ripetere il tentativo di scrittura, dopo aver spostato la linguetta.

Come si è visto, il programma prevede di chiamare IoErr e di controllare i valori della variabile EN ogni volta che viene generato un nuovo valore di EN in una della quindici routine. Abbiamo scelto questa via a scopo didattico, per aiutarvi ad associare gli errori e i loro nomi simbolici alla particolare operazione che il programma cerca di eseguire.

C'è però una tecnica più diretta per riconoscere ed elaborare questi errori, che consente di abbreviare notevolmente il programma. Si può infatti riscrivere la funzione (o le funzioni) LogError in modo tale che elabori tutti gli errori in un punto solo, direttamente nei suoi stessi codici. Per farlo si può lasciare EN come variabile globale, e richiedere che LogError riconosca tutti i valori che questa variabile può assumere a mano a mano che vengono chiamate le quindici routine; oppure si può riscrivere LogError aggiungendo la variabile EN tra gli argomenti che devono esserle passati, ed eseguire il controllo dell'argomento nel suo stesso svolgimento.

In entrambi i casi il programma diventerà notevolmente più breve, e le operazioni di elaborazione degli errori saranno concentrate tutte nello stesso punto. Realizzare questo cambiamento potrebbe essere un utile esercizio di programmazione.

Nel prossimo numero

In questo articolo abbiamo discusso la gestione di una buona metà delle funzioni della libreria DOS/ARP relative a file e directory. Nel prossimo ci concentreremo su un programma che illustri le altre: Input, Output, WaitForChar, Lock, UnLock, DupLock, ParentDir, Examine, ExNext, GetPacket, QueuePacket, DateStamp ed Execute.

Il programma che sarà associato all'articolo, in coppia con quello proposto in queste pagine, servirà come modello per la costruzione delle sezioni di programma relative alla gestione di file e directory.

Listato: *Funzioni DOS 1*

```
#include <stdio.h>
#include <exec types.h>
include <libraries dos.h>
#include <libraries dosextens.h>
#include <arpbase h>
#include <arpfunc h>

#define ERROR_NO_ARP_LIBRARY 100L

 * Abbreviamo i nomi degli errori per avere
   righe di programma piu' brevi *

#define E_NAL   ERROR_NO_ARP_LIBRARY
#define E_NADD  ERROR_NOT_A_DOS_DISK
#define E_ANK   ERROR_ACTION_NOT_KNOWN
#define E_BSN   ERROR_BAD_STREAM_NAME
#define E_CTB   ERROR_COMMENT_TOO_BIG
#define E_DNE   ERROR_DIRECTORY_NOT_EMPTY
#define E_DNF   ERROR_DIR_NOT_FOUND
#define E_DNV   ERROR_DISK_NOT_VALIDATED
#define E_DWP   ERROR_DISK_WRITE_PROTECTED
#define E_NME   ERROR_NO_MORE_ENTRIES
#define E_RAD   ERROR_RENAME_ACROSS_DEVICES
#define E_TML   ERROR_TOO_MANY_LEVELS
#define E_DF    ERROR_DISK_FULL
#define E_DP    ERROR_DELETE_PROTECTED
#define E_IL    ERROR_INVALID_LOCK
#define E_ND    ERROR_NO_DISK
#define E_OE    ERROR_OBJECT_EXISTS
#define E_SE    ERROR_SEEK_ERROR
#define E_RP    ERROR_READ_PROTECTED
#define E_WP    ERROR_WRITE_PROTECTED

char *dirName[] =
{
"DF1 Dir1", "DF1:Dir2", "DF1:Dir3"
0
}

char *newDirName[] =
{
"DF1:Dir11" "DF1 Dir22", "DF1:Dir33"
0
}

char *dirComment[] =
{
"MyDir1", "MyDir2", "MyDir3"
0
}

LONG DirProtMode = FIBF_DELETE; * Bit 1
   impostato: nessuna directory e' cancellabile *

char *fileName[] =
{
"DF1 Dir1 File1" "DF1:Dir1 File2"
   "DF1:Dir1 File3"
"DF1:Dir2 File4", "DF1 Dir2 File5"
   "DF1 Dir2 File6"
"DF1 Dir3 File7" "DF1 Dir3 File8"
   "DF1:Dir3 File9"
0
}

char *newFileName[] =
{
"DF1:Dir1/File11", "DF1 Dir1 File22"
   "DF1:Dir1 File33"
"DF1:Dir2 File44" "DF1 Dir2 File55"
   "DF1:Dir2 File66"
"DF1:Dir3 File77" "DF1 Dir3 File88"
   "DF1 Dir3 File99"
0
};

char *fileText[] =
{
"Questo e' il testo di file 1\n" "Questo e'
   il testo di file 2\n" "Questo e' il testo
   di file 3\n"
"Questo e' il testo di file 4\n" "Questo e'
   il testo di file 5\n" "Questo e' il testo
   di file 6\n"
"Questo e' il testo di file 7\n" "Questo e'
   il testo di file 8\n" "Questo e' il testo
   di file 9\n"
0
}

char *newFileText[] =
{
```

```
"Testo aggiunto a file 1", "Testo aggiunto
   a file 2", "Testo aggiunto a file 3",
"Testo aggiunto a file 4", "Testo aggiunto a file
   5", "Testo aggiunto a file 6",
"Testo aggiunto a file 7", "Testo aggiunto a file
   8", "Testo aggiunto a file 9",
0
};

char *fileComment[] =
{
"File 1", "File 2", "File 3",
"File 4", "File 5", "File 6",
"File 7", "File 8", "File 9",
0
};

LONG FileProtMode = FIBF_DELETE; /* Bit 1
   impostato: nessun file e' cancellabile */

char *readFileText[9][25];

struct ArpBase *OpenLibrary();
struct ArpBase *ArpBase;
struct FileLock *dirLock[3];
struct FileLock *oldDirLock[3];

struct FileLock *origDirLock;
struct FileLock *dummyDirLock;

struct FileLock *fileLock[9];

struct FileHandle *fileHandle[9];

LONG EN;

SHORT i, j, OF;
BOOL success;
FILE *fP;

/* ****************************************** */

main(argc, argv)
int argc;
char **argv;
{
  char *logFileName = "File di riepilogo";

  fP = fopen(logFileName, "a+");
  if(fP == NULL)
  {
    printf("ERRORE: il vostro file di riepilogo
       non puo' essere aperto\n", logFileName);
    exit(10L);
  }

  if(!(ArpBase = OpenLibrary(ArpName,
     ArpVersion)))
  {
    EN = E_NAL;
    LogError("Non e' possibile aprire la libreria
       ARP\n", NULL, fP);
  }

  origDirLock = CurrentDir(0L);

  EN = 0; MakeDir();
  EN = 0; OpenFile();
  EN = 0; WriteFileText();
  EN = 0; ReadFileText();
  EN = 0; SeekEOF();
  EN = 0; AddNewFileText();
  EN = 0; SetDirComment();
  EN = 0; SetFileComment();
  EN = 0; SetDirProt();
  EN = 0; SetFileProt();
  EN = 0; TryDeleteDir();
  EN = 0; TryDeleteFile();
  EN = 0; TryRenameDir();
  EN = 0; TryRenameFile();
  exit(0L);
}

/* ****************************************** */

MakeDir()
{

  for(i = 0; dirName[i] != NULL; i++)
  {
    dirLock[i] = CreateDir(dirName[i]);
    if(dirLock[i] == 0)
    {
      EN = IoErr();
      if(EN == E_ND)   LogError("Non c'e' nessun
         disco nel drive\n", NULL, fP);
      if(EN == E_DNV)  LogError("Il disco non e'
         validated\n", NULL, fP);
      if(EN == E_NADD) LogError("Il disco non e'
         DOS\n", NULL, fP);
      if(EN == E_DWP)  LogError("Il disco e'
         protetto in scrittura\n", NULL, fP);
      if(EN == E_DF)   LogError("Il disco e'
         pieno, non e' possibile creare
         la directory\n", NULL, fP);
      if(EN == E_TML)  LogError("La directory
         ha troppi livelli\n", NULL, fP);
    }
    printf("La directory %s e' stata creata\n"
       dirName[i]);
  }
}

/* ****************************************** */

OpenFile()
{

  OF = 0;
  for(j = 0; j < 9; j++)
  {
    fileHandle[j] = Open(fileName[j], MODE_NEWFILE);

    if(fileHandle[j] == 0)
    {
      EN = IoErr();
      if(EN == E_NADD) LogError("Il disco non e'
         DOS\n", NULL, fP);
      if(EN == E_DWP)  LogError("Il disco e'
         protetto in scrittura\n", NULL, fP);
      if(EN == E_DF)   LogError("Il disco e'
         pieno, non e' possibile aprire
         il file\n", NULL, fP);
      if(EN == E_OE )  LogError("Il file esiste
         gia'\n", NULL, fP);
      if(EN == E_BSN)  LogError("Il nome
         del file non e' stato specificato
         nel modo corretto\n", NULL, fP);
    }
    printf("Il file %s e' stato aperto\n",
       fileName[j]);
  }
  OF = 1;
}

/* ****************************************** */

WriteFileText()
{
  SHORT len, actuallength;

  for(i = 0; i < 9; i++)
  {
    len = strlen(fileText[i]);
    actuallength = Write(fileHandle[i]
       fileText[i] len);
    if(actuallength == -1)
    {
      EN = IoErr();
      if(EN == E_ND)   LogError("Non c'e' nessun
         disco nel disk drive\n", NULL, fP);
      if(EN == E_DNV)  LogError("Il disco non e'
         validated\n", NULL, fP);
      if(EN == E_NADD) LogError("Il disco non e'
         DOS\n", NULL, fP);
      if(EN == E_DWP)  LogError("Il disco e
         protetto in scrittura\n" NULL, fP);
      if(EN == E_WP )  LogError("Il file e'
         protetto in scrittura\n" NULL, fP);
      if(EN == E_DF)   LogError("Il disco e
         pieno, non e' possibile scrivere dati
         nel file\n", NULL, fP);
    }
    printf("Il testo del buffer e' stato scritto
       nel file %s\n",
       fileName[i])
    printf("Il testo del buffer e' %s\n'
       fileText[i])
```

```
    }
}

/* ****************************************** */

ReadFileText()
{
  SHORT len, actuallength;

  for(i = 0; i < 9; i++)
  {
    len = strlen(fileText[i]);
    actuallength = Read(fileHandle[i],
       readFileText[i], len);
    if(actuallength == -1)
    {
      EN = IoErr();
      if(EN == E_ND)    LogError("Non c'e' nessun
         disco nel disk drive\n", NULL, fP);
      if(EN == E_DNV)  LogError("Il disco non e'
         validated\n", NULL, fP);
      if(EN == E_NADD) LogError("Il disco non e'
         DOS\n", NULL, fP);
      if(EN == E_RP )  LogError("Il file e'
         protetto in lettura\n", NULL, fP);
    }
    printf("Il testo del file e' %s\n",
      readFileText[i]);
  }
}

/* ****************************************** */

SetDirComment()
{
  for(i = 0; i < 3; i++)
  {
    success = SetComment(dirName[i], dirComment[i]);

    if(success == FALSE)
    {
      EN = IoErr();
      if(EN == E_ND)    LogError("Non c'e' nessun
         disco nel disk drive\n", NULL, fP);
      if(EN == E_DNV)  LogError("Il disco non e'
         validated\n", NULL, fP);
      if(EN == E_NADD) LogError("Il disco non e'
         DOS\n", NULL, fP);
      if(EN == E_DWP)  LogError("Il disco e'
         protetto in scrittura\n", NULL, fP);
      if(EN == E_WP)   LogError("La directory e'
         protetta in scrittura\n", NULL, fP);
      if(EN == E_CTB)  LogError("Il commento e'
         troppo lungo\n", NULL, fP);
    }
    printf("Il nuovo commento e' stato inserito
       nella directory %s\n", dirName[i]);
  }
}

/* ****************************************** */

SetFileComment()
{
  for(i = 0; i < 9; i++)
  {
    success = SetComment(fileName[i], fileComment[i]
);
    if(success == FALSE)
    {
      EN = IoErr();
      if(EN == E_ND)    LogError("Non c'e' nessun
         disco nel disk drive\n", NULL, fP);
      if(EN == E_DNV)  LogError("Il disco non e'
         validated\n", NULL, fP);
      if(EN == E_NADD) LogError("Il disco non e'
         DOS\n", NULL, fP);
      if(EN == E_DWP)  LogError("Il disco e'
         protetto in scrittura\n", NULL, fP);
      if(EN == E_WP)   LogError("Il file e'
         protetto in scrittura\n", NULL, fP);
      if(EN == E_CTB)  LogError("Il commento e'
         troppo lungo\n", NULL, fP);
    }
    printf("Il nuovo commento e' stato inserito
       nel file %s\n", fileName[i]);
  }
}

/* ****************************************** */

SetDirProt()
{
  for(i = 0; i < 3; i++)
  {
    success = SetProtection(dirName[i], DirProtMode
    );
    if(success == FALSE)
    {
      EN = IoErr();
      if(EN == E_ND)    LogError("Non c'e' nessun
         disco nel disk drive\n", NULL, fP);
      if(EN == E_DNV)  LogError("Il disco non e'
         validated\n", NULL, fP);
      if(EN == E_NADD) LogError("Il disco non e'
         DOS\n", NULL, fP);
      if(EN == E_DWP)  LogError("Il disco e'
         protetto in scrittura\n", NULL, fP);
      if(EN == E_WP)   LogError("La directory e'
         protetta in scrittura\n", NULL, fP);
    }
    printf("La directory %s e' ora protetta contro
       la cancellazione\n", dirName[i]);
  }
}

/* ****************************************** */

SetFileProt()
{
  for(i = 0; i < 9; i++)
  {
    success = SetProtection(fileName[i],
       FileProtMode);
    if(success == FALSE)
    {
      EN = IoErr();
      if(EN == E_ND)    LogError("Non c'e' nessun
         disco nel disk drive\n", NULL, fP);
      if(EN == E_DNV)  LogError("Il disco non e'
         validated\n", NULL, fP);
      if(EN == E_NADD) LogError("Il disco non e'
         DOS\n", NULL, fP);
      if(EN == E_DWP)  LogError("Il disco e'
         protetto in scrittura\n", NULL, fP);
      if(EN == E_WP)   LogError("Il file e'
         protetto in scrittura\n", NULL, fP);
    }
    printf("Il file %s e' ora protetto contro
       la cancellazione\n", fileName[i]);
  }
}

/* ****************************************** */

TryDeleteDir()
{
  for(i = 0; i < 3; i++)
  {
    success = DeleteFile(dirName[i]);
    if(success == FALSE)
    {
      EN = IoErr();
      if(EN == E_ND)    LogError("Non c'e' nessun
         disco nel disk drive\n", NULL, fP);
      if(EN == E_DNV)  LogError("Il disco non e'
         validated\n", NULL, fP);
      if(EN == E_NADD) LogError("Il disco non e'
         DOS\n", NULL, fP);
      if(EN == E_DNF)  LogError("La directory
         non e' rintracciabile\n", NULL, fP);
      if(EN == E_DNE)  LogError("La directory
         non e' vuota\n", NULL, fP);
      if(EN == E_DP)   LogError("La directory e'
         protetta contro la cancellazione\n",
         NULL, fP);
      if(EN == E_DWP)  LogError("Il disco e'
         protetto in scrittura\n", NULL, fP);
    }
    printf("La directory %s e' stata
       cancellata\n", dirName[i]);
  }
}

/* ****************************************** */

TryDeleteFile()
{
```

```
for(i = 0; i < 9; i++)
{
  success = DeleteFile(fileName[i]);
  if(success == FALSE)
  {
    EN = IoErr();
    if(EN == E_ND)    LogError("Non c'e' nessun
       disco nel disk drive\n", NULL, fP);
    if(EN == E_DNV)   LogError("Il disco non e'
       validated\n", NULL, fP);
    if(EN == E_NADD)  LogError("Il disco non e'
       DOS\n", NULL, fP);
    if(EN == E_DNF)   LogError("La directory non
       e' rintracciabile\n", NULL, fP);
    if(EN == E_DWP)   LogError("Il disco e'
       protetto in scrittura\n", NULL, fP);
    if(EN == E_DP)    LogError("Il file e'
       protetto contro la cancellazione\n",
       NULL, fP);
  }
  printf("Il file %s e' stato cancellato\n",
     fileName[i]);
 }
}

/* ****************************************** */

TryRenameDir()
{
  for(i = 0; i < 3; i++)
  {
    success = Rename(dirName[i], newDirName[i]);
    if(success == FALSE)
    {
      EN = IoErr();
      if(EN == E_ND)    LogError("Non c'e' nessun
         disco nel disk drive\n", NULL, fP);
      if(EN == E_DNV)   LogError("Il disco non e'
         validated\n", NULL, fP);
      if(EN == E_NADD)  LogError("Il disco non e'
         DOS\n", NULL, fP);
      if(EN == E_RAD)   LogError("Non e'
         possibile rinominare le directory da un
         dispositivo all'altro\n", NULL, fP);
      if(EN == E_DNF)   LogError("La directory
         non e' rintracciabile\n", NULL, fP);
      if(EN == E_DWP)   LogError("Il disco e'
         protetto in scrittura\n", NULL, fP);
      if(EN == E_WP)    LogError("La directory e'
         protetta in scrittura\n", NULL, fP);
    }
    printf("Il nome della directory %s e' stato
       cambiato\n", dirName[i]);
  }
}
/* ****************************************** */

TryRenameFile()
{
  for(i = 0; i < 9; i++)
  {
    success = Rename(fileName[i], newFileName[i]);
    if(success == FALSE)
    {
      EN = IoErr();
      if(EN == E_ND)    LogError("Non c'e' nessun
         disco nel disk drive\n", NULL, fP);
      if(EN == E_DNV)   LogError("Il disco non e'
         validated\n", NULL, fP);
      if(EN == E_NADD)  LogError("Il disco non e'
         DOS\n", NULL, fP);
      if(EN == E_RAD)   LogError("Non e' possibile
         rinominare un file da un dispositivo
         all'altro\n", NULL, fP);
      if(EN == E_DNF)   LogError("La directory
         non e' rintracciabile\n", NULL, fP);
      if(EN == E_DWP)   LogError("Il disco e'
         protetto in scrittura\n", NULL, fP);
      if(EN == E_WP)    LogError("Il file e'
         protetto in scrittura\n", NULL, fP);
    }
    printf("Il nome del file %s e' stato
       cambiato\n", fileName[i]);
  }
}

/* ****************************************** */

[illegible]

{
int oldposition;

  for(i = 0; i < 9; i++)
  {
    oldposition = Seek(fileHandle[i], 0,
       OFFSET_END);
    if(oldposition == -1)
    {
      if(EN == E_ND)    LogError("Non c'e' nessun
         disco nel disk drive\n", NULL, fP);
      if(EN == E_DNV)   LogError("Il disco non e'
         validated\n", NULL, fP);
      if(EN == E_NADD)  LogError("Il disco non e'
         DOS\n", NULL, fP);
      if(EN == E_SE)    LogError("Il disco ha
         incontrato un errore fisico
         nella ricerca\n", NULL, fP);
    }
    printf("Ora la posizione del cursore nel
       file %s e' quella finale\n", fileName[i]);
  }
}

/* ****************************************** */

AddNewFileText()
{
  SHORT len, actuallength;

  for(i = 0; i < 9; i++)
  {
    len = strlen(newFileText[i]);
    actuallength = Write(fileHandle[i],
       newFileText[i], len);
    if(actuallength == -1)
    {
      EN = IoErr();
      if(EN == E_ND)    LogError("Non c'e' nessun
         disco nel disk drive\n", NULL, fP);
      if(EN == E_DNV)   LogError("Il disco non e'
         convalidato\n", NULL, fP);
      if(EN == E_NADD)  LogError("Il disco non e'
         DOS\n", NULL, fP);
      if(EN == E_DWP)   LogError("Il disco e'
         protetto in scrittura\n", NULL, fP);
      if(EN == E_WP)    LogError("Il file e'
         protetto in scrittura\n", NULL, fP);
      if(EN == E_DF)    LogError("Il disco e'
         pieno, non e' possibile creare
         una nuova directory\n", NULL, fP);
    }
    printf("Il nuovo testo proveniente dal buffer
       e' stato scritto nel file %s\n",
       fileName[i]);
  }
}

/* ****************************************** */

CloseFile()
{
  for(i = 0; i < 9; i++)
  {
    Close(fileHandle[i]);
  }
  printf("Tutti e nove i file sono stati
     chiusi\n", fP);
}

/* ****************************************** */

LogError(errorStr, infoStr, fP)
char *errorStr;
char *infoStr;
FILE *fP;
{
  fputs(errorStr, fP);
  if(infoStr != NULL) fputs(infoStr, fP);

  if(EN > 0)
  {
    if(fP != NULL) fclose(fP);
    if(OF == 1) CloseFile();
    if(ArpBase != NULL) CloseLibrary(ArpBase);
    printf("Siete usciti dal programma
       prematuramente\n");
    dummyDirLock = CurrentDir(origDirLock);
    exit(10);
  }
}
```

GENIAS, PER UN NUOVO SOFTWARE ITALIANO

Commodore Gazette è andata a curiosare "dietro le quinte" di una nuova software house, specializzata nella produzione di videogiochi

di Fabio Rossi

Un guerriero armato di ascia si aggira fra pietre tombali rovesciate e giganteschi alberi piegati in sagome innaturali e inquietanti: la sua destinazione è il futuro del software italiano. Il nostro eroe debutterà in *Dragon's Kingdom*, prossima produzione della software house più giovane d'Italia, e di lui si sa soltanto che si è già reincarnato più di dieci volte e che il suo viaggio è cominciato a Castenaso, un paesino nei pressi di Bologna.

Qui è cominciato anche il nostro più tranquillo viaggio all'interno del software *made in Italy*. A Castenaso si trova infatti la sede della Genias, fondata da Riccardo Arioti come divisione della Multimedia. È proprio con Riccardo che ci siamo incontrati nel periodo delle vacanze pasquali, per scoprire qualcosa di più della Genias e dei suoi progetti futuri.

D. *Partiamo dall'inizio: com'è nata la Genias?*

R. La fondazione di una software house era un mio vecchio progetto. Prima di dare il via alla Genias, in effetti, ho collaborato alla creazione della Simulmondo, altra casa di software bolognese specializzata nella produzione di giochi. Quella partecipazione è poi naufragata a causa di una serie di divergenze con uno dei soci, che basandosi su una serie di convinzioni personali era riuscito a stravolgere l'immagine della casa e a minare i rapporti con la stampa, con gli autori e con i rivenditori. Il passo successivo è stato quello della ricostruzione: ho voluto continuare l'esperienza ricominciando dall'inizio, insieme a un certo numero di persone che avevano una visione più realistica del mercato. La presentazione ufficiale del marchio Genias risale al 28 ottobre del 1989.

D. *Come descriverebbe la politica della società?*

R. Intendiamo produrre software di ogni genere, dal momento che i gusti europei sono estremamente differenziati, e desideriamo superare le frontiere italiane sin da adesso. In Inghilterra, per esempio, il genere degli shoot'em up che qui viene tanto bistrattato ha ancora un grande seguito, quindi abbiamo in lavorazione un programma di quel tipo rivolto essenzialmente al mercato estero.

D. *Parliamo dei prodotti già usciti.*

R. Per il momento abbiamo pubblicato un'avventura semigrafica intitolata *Mystere*, e un gioco ispirato ai Mondiali di calcio che si chiama *World Cup 90*.

D. Mystere *non è stato accolto con troppo favore.*

R. Questo introduce una nuova importante tematica: l'incentivazione del settore. Con *Mystere* volevamo raggiungere due scopi: pubblicare in tempi brevi un'avventura in italiano, cioè un tipo di programma molto richiesto ma purtroppo praticamente scomparso nel nostro Paese, e dimostrare che la Genias può dare grandi soddisfazioni ai programmatori italiani. La versione per Amiga è stata una specie di esperimento, e la conversione per C-64 dovrebbe comunque essere molto migliore.

D. World Cup 90 *invece è un ottimo programma, e sta ottenendo un meritato successo.*

R. Nell'anno dei Mondiali in Italia non potevamo farci scappare l'occasione di pubblicare un gioco di calcio. Nella fase di realizzazione abbiamo tenuto conto della terribile concorrenza degli inglesi, e abbiamo dimostrato che con un'attenta progettazione si possono creare giochi estremamente concorrenziali.

D. *Il che ci porta ai rapporti fra la Genias e le software house straniere.*

R. Al recente ECPS '90, in Inghilterra, abbiamo concluso alcuni contratti che assicurano la distribuzione dei nostri giochi in tutta Europa, in Australia e – solo per le versioni PC – nelle Americhe. Abbiamo anche avuto la soddisfazione di essere stati prescelti dal gruppo NEC, fra una cinquantina di case, per la conversione di titoli europei sulle console giapponesi. A questo punto, i rapporti con la concorrenza estera non possono che essere di parità.

D. *I primi giochi che avete prodotto sono usciti in versione Amiga e C-64/128. In futuro avete intenzione di sviluppare software anche per altre macchine?*

R. In Europa esiste un parco macchine molto vario: in Spagna ci sono molti utenti Spectrum, in tutto l'occidente c'è una forte presenza della gamma Amstrad e così via. Vorremmo riuscire a convertire i giochi in tutti i formati possibili, ma il grosso problema è che in Italia certe macchine sono del tutto sconosciute ai programmatori. Il primo computer per il quale si sviluppa, comunque, è sempre l'Amiga.

D. *Che ne pensate della morte annunciata dei computer a 8 bit, e in particolare del C-64?*

R. Abbiamo un quadro molto chiaro di quella che sarà la situazione nei prossimi due anni. Nella sola Italia sono stati venduti tre milioni di C-64, che corrispondono a una possibile vendita di almeno 30 mila copie per ogni gioco. Gli altri 8 bit sono scomparsi perché mal supportati dal software, ma il C-64 resisterà almeno fino al '92, nonostante l'avanzata massiccia dell'Amiga. Il problema più grosso, in questo campo, è dovuto al fatto che per molte software house produrre giochi e programmi per il C-64 risulta antieconomico, dal momento che con queste macchine i margini di profitto sono drasticamen-

te ridotti.

D. *Quindi i "sessantaquattristi" italiani possono contare sulla Genias?*

R. Senz'altro, e per due buone ragioni: la prima è che questo mercato finirà per offrire sempre meno concorrenza, e la seconda è che con i computer a 16 bit come l'Amiga la pirateria è ancora un problema e quindi frena lo sviluppo.

D. *Una piccola domanda "en passant": che cosa vuol dire "Genias"?*

R. (Ridendo) Nulla! All'inizio volevamo chiamarla "Genesis", ma ci saremmo confusi con gruppi musicali, console giapponesi, motociclette... Dopo qualche rimaneggiamento è venuto fuori questo nome, che ricorda anche la parola "genio", nel senso di creatività.

D. *Tornando al software, quali sono i progetti per il futuro?*

R. Molti. Quelli di cui posso parlare sono un'avventura, sviluppata da un gruppo di veri appassionati, dove il protagonista viene rapito per errore e finisce in mezzo a un sacco di guai. Poi c'è quel *Dragon's Kingdom* che hai già visto, un'avventura dinamica che punta molto sulla suggestiva atmosfera creata da grafica e sonoro. E ancora c'è un gioco automobilistico intitolato *Warm-Up*, ambientato nel mondo della Formula Uno... e tanti altri progetti, come per esempio i programmi educativi.

D. *Nessun'altra anticipazione?*

R. Un nostro collaboratore sta lavorando a un gioco per Amiga intitolato *Deux*, un'avventura dinamica davvero eccezionale. Purtroppo l'autore è un terribile perfezionista, e ogni volta che scopre una nuova routine vuole a tutti i costi inserirla, anche se bisogna riscrivere il programma daccapo. *Deux* sarebbe potuto uscire già parecchi mesi fa, e avrebbe fatto un'ottima figura, ma lui non era soddisfatto. Ormai non so più dire quando uscirà, ma per allora sarà qualcosa di unico.

D. *So che avete anche un team dedicato soltanto alla realizzazione di avventure grafiche, vero?*

R. Vero. Stiamo lavorando a un editor per avventure grafiche simile a quelli della Sierra o della Lucasfilm, capace di gestire immagini HAM senza sfarfallii. Entro Natale speriamo di pubblicare il primo titolo realizzato con questo sistema.

D. *La Genias si sta muovendo anche in altre direzioni, non è vero?*

R. Stiamo acquisendo i diritti di pubblicazione di alcuni programmi d'indubbio valore, che sono rimasti bloccati nei cassetti di alcune software house per diverse ragioni. Presto dovrebbe uscire *Over the Net*, un [bellissimo, *ndr*] gioco di beach-volley realizzato dai fratelli Dardari, gli autori di *World Cup 90* versione Amiga. Stiamo anche cercando di ottenere i diritti di *Crimetown Depths* (vedi *Commodore Gazette* numero 3/89), un gioco italiano dalla gestazione molto, molto lunga e molto difficile.

D. *Quanti titoli prevedete di pubblicare nei prossimi mesi?*

R. Per il 1990 almeno sei, e dall'anno prossimo cercheremo di tenere una media di dieci titoli all'anno.

D. *Hai parlato di software didattico. Ma è una strada che già altre case hanno tentato di battere, e nessuna ha avuto un grande successo.*

R. Quello che abbiamo intenzione di realizzare non è rivolto alle istituzioni scolastiche: al momento stiamo per completare il primo di una serie di corsi di programmazione. Si tratta di un corso di BASIC per PC, che a differenza di tanti prodotti simili si svolge integralmente a video e offre una notevole interattività con l'utente. La confezione conterrà tre dischi da 3,5"e tre da 5,25": alla fine del corso uno dei dischi conterrà anche un programma realizzato interamente dall'utente. Oltre all'interattività, la caratteristica distintiva di questo programma sarà il prezzo: qualche tempo fa la IBM aveva realizzato qualcosa di simile, ma la vendeva al prezzo proibitivo di un milione e 400 mila lire. Noi pensiamo di vendere il nostro pacchetto a circa 250 mila lire.

D. *La Genias ha in progetto particolari iniziative promozionali, per il futuro?*

R. Almeno due, di cui una è purtroppo top secret. Quella che possiamo rivelare riguarda un torneo di *World Cup 90* che si svolgerà verso la fine dell'estate e che sfrutterà le capacità della nostra interfaccia per quattro giocatori.

D. *Per concludere, come presenteresti la Genias?*

R. Questa software house non è un'avventura, un tentativo alla cieca, ma è un serio sforzo di competizione con le case estere. Fino a che i programmatori italiani non si renderanno conto che hanno la possibilità di lavorare in patria a livello professionale, l'industria del software italiana opererà sempre a una frazione delle sue potenzialità. ■

IL TEAM GENIAS

Con questo breve elenco di nomi vogliamo rendere un piccolo omaggio a un gruppo di persone che cercano di tenere alto l'onore nazionale nel campo del software ludico. Sono gli autori attualmente al lavoro per la Genias.

World Cup 90

(versione Amiga)
Davide Dardari (23 anni)
Francesco Dardari (18)
Marco Dardari (17)

(versione C-64/128)
Antonio Miscellaneo (17)
Ivan Del Duca (18)

Dragon's Kingdom

(versione Amiga)

Massimiliano Agostinelli (20)	Codice
Francesco Martire (18)	Codice
Nicola Tomilianovich (17)	Effetti sonori
Fabio Rossi (semplice omonimia, *ndr*)	Grafica
Emanuele Tajariol (20)	Grafica
Raffaele Valansise (20)	

(versione C-64/128)
Alberto Frabetti (24)

Team "Adventure"

Giuseppe Orofino (22)	Codice
Gianluca Orofino (20)	Musica
Gabriele Gabrielli (19)	Codice
Alessandro Marras (19)	Grafica e sceneggiatura
Cesare Manzini (20)	Grafica e sceneggiatura
Lorenzo Ranuzzi (20)	Grafica e sceneggiatura
Fabrizio Macaluso	conversione Atari ST
Andrea Paselli	conversione C-64
Luca Zarri	conversione C-64

Fra i musicisti dovremmo inserire anche Stefan Roda (23), che però prima di tutto ricopre il ruolo di software manager nella sede centrale. Ringraziamo Stefan per la collaborazione offerta durante il breve soggiorno bolognese

Mentre andiamo in stampa, sono in corso le trattative per la formazione di un gruppo "staccato" in Lombardia, che dovrebbe svolgere incarichi particolari. . ma è tutto ancora assolutamente top secret

La Genias è molto interessata a contattare nuovi programmatori, grafici e musicisti. Chi desidera mettersi in contatto può farlo scrivendo a. Genias, Via Bentivogli 56, 40055 Castenaso (BO)

OGNI PEZZO AL SUO POSTO: CONNEX!

Colpo d'occhio e rapidità sono le qualità per avere successo in questo gioco ispirato al famoso Tetris, ma dotato di un'innegabile originalità

di Leonard Morris

Vi va di passare un po' di tempo con un gioco d'incastri? Allora fate conoscenza con *Connex*, un gioco per 128 che ricorda per qualche aspetto il famoso *Tetris*. Scopo del gioco è incastrare l'uno all'altro "blocchetti" di cinque quadratini generati casualmente; si deve ovviamente riempire il più possibile il rettangolo di gioco.

Nel caso dell'opzione per due giocatori, il programma consente la scelta tra un solo campo di gioco oppure due separati. I giocatori 1 e 2 usano joystick collegati rispettivamente alla porta 1 e alla porta 2.

Quando appare il primo blocchetto sulla parte alta dello schermo, spostatelo sulla tavola di gioco in modo che tocchi l'unico quadratino già presente; poi, prima che scatti il limite di tempo, premete il pulsante Fire del joystick per fissarlo. I punti che si guadagnano sono proporzionali al numero di quadratini che toccano quello già presente. Se il blocchetto non tocca il quadratino, vi si sovrappone o esce parzialmente dalla tavola di gioco, oppure se superate il tempo massimo, venite penalizzati perdendo il blocchetto e il turno di gioco.

Quando appare il secondo blocchetto, dovete unirlo al primo e così via, riempiendo il maggior numero possibile di quadratini sulla tavola. A differenza di *Tetris*, questo gioco permette di spostare i blocchetti anche attraverso quelli già collocati, ma non permette di ruotarli. Dopo quattro penalità il gioco finisce e dovete ricominciare daccapo. Se però riuscite a riempire il 75 per cento della tavola, avanzate al successivo livello di gioco.

Un altro modo per avanzare al livello successivo (ricevendo anche un bonus di 1000 punti) è quello di riempire completamente una riga: la riga svanisce e quelle superiori scendono riempiendo lo spazio lasciato libero.

Connex ha 16 livelli, ciascuno reso più difficile del precedente da un crescente numero di cerchietti "predatori" che vagano sul terreno di gioco. Se uno dei cerchietti tocca il blocchetto che state muovendo, lo perdete e vi viene assegnata una penalità. Quando si gioca in due, il diverso colore dei blocchetti indica a chi tocca. Se il blocchetto è giallo è il turno del primo giocatore, se è rosso tocca al secondo. Benché vi sia un limite di tempo per ogni mossa, ho preferito non visualizzare un cronometro sullo schermo... un po' per aumentare la suspense e un po' perché l'azione è comunque troppo veloce per consultarlo.

Durante il gioco, sullo schermo sono sempre presenti tre numeri: il punteggio che avete raggiunto, la situazione delle penalità e la percentuale della tavola di gioco coperta fino a quel momento. Assicuratevi di aver salvato sul disco il programma prima di dare il RUN. Oltre a questo non ci sono altri commenti da fare. Buoni incastri!

Istruzioni: ***digitare il listato, salvarlo su disco e impartire il run. Tutti i caratteri grafici e di controllo sono stati tradotti in combinazioni di tasti leggibili. Sono le istruzioni tra parentesi graffe. Per ulteriori informazioni vedere "Come digitare i listati" a pagina 94.***

```
0,00,BD,D6,11,DD,10,0B,F0,0F,10,08,F
E
88 DATA D6,11,A0,01,4C,E8,13,DE,D6,1
1,A0,01,BD,D7,11,DD,11,0B,F0,0F,10,0
8
89 DATA FE,D7,11,A0,01,4C,FF,13,DE,D
7,11 A0,01,E8,E8,E0,0A,D0,CC,A9,02,8
5
90 DATA FE,A9,00,85,FD,C6,FD,D0,FC,C
6,FE,D0,F4,98 D0,E5,60,A0,00,84,FA,A
9
91 DATA 00,85 FB,A5,FA,0A,0A,18,65,F
B,AA,BD,00,0B,F0,18,A5,FE,0A,0A,0A,1
8
92 DATA 69,80,99,10,0B,A5,FA,0A,0A,0
A,18,69,32,99,11,0B,C8,C8,E6,FB,A5,F
B
93 DATA C9,04,D0,D3,E6,FA,A5,FA,C9,0
4,D0,C7,20,CD,13,60,A2,00,BD,D6,11,3
8
94 DATA E9,18,4A,4A,4A,85,FA,BD,D7,1
1,38,E9,32,4A,4A,4A,85,FB,A9,04,85,F
D
95 DATA A5,FA,85,FC,A5,FB,F0,10,A5,F
C,18,69,28,85,FC,90,02,E6,FD,C6,FB,4
C
96 DATA 7A,14,A0 00,B1,FC,C9,20,4C,2
D,13,A5 FC,9D,30,0B,A5,FD,9D,31,0B,E
8
97 DATA E8 E0 0A,D0,B5,4C,DA,14,A5,F
C,38,E9 28 85 FC,B0,02,C6,FD,B1,FC,C
9
99 DATA D1,60,A5,FC,18,69,27,85,FC,9
0,02 E6 FD B1,FC C9,D1,60 A5,FC,18,6
9
99 DATA 02 85 FC,90 02,E6,FD,B1,FC,C
9 D1 60,A0 00,84 FA,A2,00 BD,30,0B,8
5
100 DATA FC BD,31 0B,85,FD,20,AA,14,
D0,02 E6 FA 20,BA,14,D0,02,E6,FA,20,
CA
101 DATA 14,D0,02,E6 FA,20,BA 14,D0,
02,E6 FA,E8,E8,E0 0A,D0,D4,A5,FA,D0,
03
102 DATA 4C,2F 13,A2,00,BD,30,08,85,
FC BD 31,0B 85,FD,A9,D1 91,FC,E8,E8,
E0
103 DATA 0A,D0 EC,60,AD,07,DC,29,0F
0A,0A,0A 0A,8D,48,13,8D,75,13,8D,A9,
13
104 DATA 8D BE,13,EE 2F,0B,AD 2F 0B,
29 03 D0 07,4C 45,13 C9,01,D0,03,4C
56
105 DATA 13 C9,02 D0,03 4C,84,13 4C
B9 13 A9,00 8D FE,19 A0,00 B9 00 1A
8D
106 DATA 74,15 B9 01,1A,8D 73,15,A2,
00 BD 72 07 C9 20,F0 03 E8,E0 19 D0
F4
107 DATA 20 B2 15,C8 C8 C0 20 D0 DD
60 A0 00 B9 00 1A 8D,9C 15 B9,01 1A
8D
108 DATA 9B 15 A2 00 A9 20 9D 72 07,
E8 E0 19 D0 F6 C8 C8 C0 20 D0 E2 60
A0
109 DATA 19 D0 F0 60 EA,EA EA,60,EE,
FE,19,98,8D,FC,19,AD,74,15,8D,E9,15,
8D
110 DATA EC,15,AD,73,15,8D,E8,15,AD,
E8,15,38,E9,28,8D,E8,15,8D,EB,15,B0,
06
111 DATA CE,E9,15,CE,EC,15,AD,EC,15,
C9,04,F0,13,A2,00,A0,28,BD,F2,04,99,
F2
112 DATA 04,C8,E8,E0,19,D0,F4,4C,C8,
15,20,EC,17,A8,60,A9,30,8D,1A,16,A9,
00
113 DATA 8D,19,16,A0,00,B9,00,1A,8D,
17,16,B9,01,1A,8D,16,16,A2,00,BD,72,
07
114 DATA 9D,80,32,E8,E0,19,D0,F5,AD,
19,16,18,69,28,8D,19,16,90,03,EE,1A,
16
115 DATA C8,C8,C0,20,D0,D3,60,A9,20,
8D,51,16,A9,00,8D,50,16,A0,00,B9,00,
1A
116 DATA 8D,54,16,B9,01,1A,8D,53,16,
A2,00,BD,80,22,9D,72,07,E8,E0,19,D0,
F5
117 DATA AD,50,16,18,69,28,8D,50,16,
90,03,EE,51,16,C8,C8,C0,20,D0,D3,60,
EA
118 DATA A2,00,A9,00,9D,00,0B,E8,E0,
10,D0,F6,A2,05,AD,06,DC,29,0F,A8,B9,
00
119 DATA 0B,D0,F5,A9,01,99,00,0B,CA,
D0,ED,4C,19,14,FF,3E,BD,E1,11,C9,62,
F0
120 DATA 03,DE,E1,11,60,BD,E1,11,C9,
E2,F0,03,FE,E1,11,60,BD,E0,11,C9,1D,
F0
121 DATA 03,DE,E0,11,60,BD,E0,11,C9,
E8,F0,03,FE,E0,11,60,EE,94,16,AD,94,
16
122 DATA F0,01,60,A9,FE,8D,94,16,EE,
95,16,AD,95,16,C9,40,D0,05,A9,3D,8D,
95
123 DATA 16,8D,FF,07,8D,FE,07,8D,FD,
07 A2,00,BD,00,38,29,07,0A,0A,0A,EA,
8D
124 DATA F7,16,20,20,17,E8,E8,E0,06,
D0,EB,60,20,96,16,4C,96,16,01,00,20,
96
125 DATA 16,4C,B7,16,EA,EA,20,B7,16,
4C,B7,16,EA,EA,20,B7,16,4C,A1,16,EA,
EA
126 DATA 20,A1,16,4C,A1,16,EA,EA,20,
A1,16,4C,AC,16,EA,EA,20,AC,16,4C,AC,
16
127 DATA EA EA,20,96,16,4C,AC,16,FF,
EA 78 A9,17,8D,15,03,A9,4D,8D,14,03,
58
128 DATA 60 20 00 18,4C 65,FA,A0,00,
8C 06 17 8C 07,17,B9,00,1A,8D,6B,17,
B9
129 DATA 01 1A,8D 6A,17 A2,00,BD,72,
07,C9,D1 D0,08,EE,06,17,D0 03 EE 07,
17
130 DATA E8 E0,19 D0,EC,C8,C8 C0,20,
D0 D8 60 A0 08 A2,00 FE D6 11,E8 E8
E0
131 DATA 0A D0,F7,88,D0 F2 EA EA EA
60,A0,08,A2,00,DE,D6,11,E8,E8,E0,0A,
D0
132 DATA F7,88,D0,F2,EA,EA,EA,60,A0,
08,A2,00,FE,D7,11,E8,E8,E0,0A,D0,F7,
88
133 DATA D0,F2,EA,EA,EA,60,A0,08,A2,
00,DE,D7,11,E8,E8,E0,0A,D0,F7,88,D0,
F2
134 DATA EA,EA,EA,60,20,84,17,4C,C0,
17,20,84,17,4C,AC,17,20,98,17,4C,AC,
17
135 DATA 20,98,17,4C,C0,17,A2,19,A9,
20,9D,19,05,CA,D0,F8,AD,FC,19,60,01,
FF
136 DATA 00,FF,00,8E,AD,FA,17,F0,0A,
AD,1E,D0,29,1F,F0,03,8D,FF,19,20,C2,
16
137 DATA 60,FF,00,A0,00,B9,20,1A,8D,
29,18,B9,21,1A,8D,28,18,A2,00,A5,FE,
9D
138 DATA 72,DB,E8,E0,19,D0,F6,C8,C8
C0,20,D0,E2,60,A0,00,B9,20,1A,8D,52,
18
139 DATA B9,21,1A,8D,51,18,A2,00,EE,
FF,1A,AD,FF,1A,29,0F,F0,F6,9D,72,DB,
E8
140 DATA E0,19,D0,EE,C8,C8,C0,20,D0,
DA,60,FF,A0,00,B9,20,1A,8D,74,18,B9,
21
141 DATA 1A,8D,73,18,A2,00,A5,FE,9D,
72,DB,E8,E0,19,D0,F6,C8,C8,C0,20,D0,
E2
142 DATA 60,E6,FE A5,FE,29,0F,F0,F8,
85,FE,4C,60,18,A2,00,A9,00,9D,00,0B,
E8
143 DATA E0,10,D0,F6,A2,05,AD,06,DC,
29,0F,C9,03,F0,F7,C9,07,F0,F3,C9,0A
10
144 DATA EF,A8,B9,00,0B,D0,E9,A9,01,
99,00,0B,CA,D0,E1,4C,19,14,A2,00,A9
00
145 DATA 9D,00,0B,E8,E0,10,D0,F6,A2,
05,AD,06,DC,29,0F,C9,05,F0,F7,C9,06,
F0
146 DATA F3,C9,09 F0,EF,C9,0A,F0 E8,
A8,B9,00 0B,D0,E5,A9,01,99,00 0B,CA,
D0
147 DATA DD,4C,19,14,00
148 DATA 0,4,8,12,13,1,4,5,6,9 0,4,8
,12,5,0,1,5,6,9,0,1,5,9,10,0,1 2,5 9
,0,1,5
149 DATA 9,8,1,2,5,6,9 0 4,8 12,9,0,
4,8 12,1,5,6,8 9,10,1,5,6 10,11,0,1,
2,6,10
150 DATA 1,2,4,5,9,1 2 5,6,10,2,6,10
,14,9,31,5418,0,63,5418,1,127,5418 2
,255
151 DATA 5418,3 31 6286,0,63 6286,1,
127,6286,2 255 6286,3 31,6334,0,63 6
334
152 DATA 1,127,6334,2,255 6334,3,255
 5744,0,127,5744,1,127,5744 2,255 57
44,3
```

■

RISCALDAMENTO DEL GLOBO? SCIOCCHEZZE!

LA PROSSIMA ERA GLACIALE COMINCERA' FRA DUE MESI

Immaginate che una meteora di 30 Kilometri si schianti sulla terra. Immaginate una nuova Era glaciale. La terra, colpita e ghiacciata ,lotta per l'ultimo angolo di salvezza! Questo è minacciato da invasori spietati. Siete in possesso delle informazioni riguardanti la storia dell'isola, delle carte e delle mappe che riguardano il vostro territorio di 80.000 miglia quadrate e di una squadra di 32 difensori. Il primo colpo è partito ed il tempo non è certo dalla vostra parte.

***Lo scopo è la sopravvivenza**
*** L'area di gioco è un territorio di 80.000 miglia quadrate**
*** Controllo su 32 uomini**
*** Un computer su terreno tridimensionale mai visto prima**
*** Manuali e mappe dettagliati.**

Quando la polvere di diamanti nell'atmosfera raggiunge livelli critici, il risultato è... una combinazione ad alta strategia ed azione. Succede solo una volta ogni 5 milioni di anni.

DISPONIBILE PER AMIGA, ATARI ST E PC

LEADER

UN NUOVO MURO DA ABBATTERE

Come vanno i vostri riflessi? Se volete metterli alla prova, ecco l'ultima variazione sul tema dei "muretti"

di Thomas Czarnecki

Si dice che Giosuè, un famoso personaggio biblico, abbia fatto crollare le mura di Gerico portando in processione per sette giorni l'Arca dell'Alleanza, accompagnata dal suono delle trombe. Anche voi avete ora la possibilità di ripetere l'impresa, abbattendo mura meno bibliche ma più colorate. *Colorout* è un gioco dal ritmo serrato (per una sola persona) che vi sfida ad abbattere il maggior numero possibile di mattoncini di un muro multicolore a sei strati.

I missili che usate sono dieci palline ad alta velocità, mantenute in gioco tramite una "racchetta" spostabile che si controlla con il joystick collegato alla porta 2, o con i tasti "," e "." della tastiera.

Il muretto è largo quanto tutto lo schermo e le palline lo colpiscono arrivando dalla parte bassa dello schermo. La racchetta si muove avanti e indietro sulla linea di fondo.

Per far partire una pallina si deve premere il pulsante Fire del joystick, oppure la barra spaziatrice della tastiera. Tenendo premuto il pulsante Fire (o la barra) si raddoppia la velocità della racchetta. Per mandare la pallina nella direzione voluta conviene tentare di darle un leggero taglio o di colpirla con l'estremità della racchetta. Se la mancate, la pallina "esce" dalla parte inferiore dello schermo. L'accompagnamento di effetti sonori aggiunge al gioco un tocco di divertimento in più.

All'inizio l'azione è lenta, ma si fa sempre più veloce a mano a mano che appaiono i nuovi muretti. La pallina abbatte soltanto il mattoncino che tocca, ma se riuscite ad aprire un varco nel muro la vedrete rimbalzare tra il "soffitto" e la parte alta del muro cancellando ogni mattoncino che tocca. Ogni mattoncino che viene abbattuto aggiunge al punteggio un valore pari al numero della fila moltiplicato per cinque. In altre parole, i mattoncini della fila più bassa valgono cinque punti, quelli della seconda dieci e così via. Nella parte alta dello schermo sono sempre presenti il punteggio raggiunto fino a quel momento, il punteggio record ottenuto nelle partite precedenti, e il numero della pallina con cui state giocando.

La partita finisce quando avete esaurito le dieci palline a vostra disposizione. Per iniziarne un'altra si deve premere il tasto Restore.

Il programma è scritto in un misto di Basic e Linguaggio Macchina. Digitatelo con attenzione e salvatelo sul disco, dopodiché date il RUN per creare *Colorout*. Quando volete iniziare il gioco, caricate quest'ultimo e date il RUN come fareste per qualunque programma Basic.

Istruzioni: digitare il listato, salvarlo su disco e impartire il RUN.
Tutti i caratteri grafici e di controllo sono stati tradotti in combinazioni di tasti leggibili. Sono le istruzioni tra parentesi graffe.
Per ulteriori informazioni vedere "Come digitare i listati" a pagina 94.

```
0 REM QUESTO LISTATO CREA IL PROGRAM
MA COLOROUT
5 OPEN 8,8,8,"COLOROUT,P,W"
6 CT=0:PRINT"{SHFT CLR}"
10 READ A$:IF A$="-1" THEN CLOSE8:PR
INT:PRINT"TUTTO OK ":END
12 PRINT"{HOME}LEGGO LA RIGA "+STR$(C
T):CT=CT+1
15 IF LEN(A$)<62 THEN 55
20 B$=MID$(A$,1,20)+MID$(A$,22,20)+
MID$(A$,43,20)
25 FOR I=1 TO 30
30 C$=MID$(B$,I*2-1,2):H$=LEFT$(C
$,1):L$=RIGHT$(C$,1)
35 H=VAL(H$):IF H$>"9" THEN H=ASC(H
$)-55
40 L=VAL(L$):IF L$>"9" THEN L=ASC(L
$)-55
45 BY=H*16+L:PRINT#8,CHR$(BY);
50 NEXT:GOTO 10
55 IF LEN(A$)<21 THEN B$=A$:GOTO 70
60 IF LEN(A$)<42 THEN B$=LEFT$(A$,20)
+RIGHT$(A$,LEN(A$)-21):GOTO 70
65 B$=LEFT$(A$,20)+MID$(A$,22,20)+R
IGHT$(A$,LEN(A$)-42)
70 FOR I=1 TO LEN(B$)/2
75 C$=MID$(B$,I*2-1,2):H$=LEFT$(C
$,1):L$=RIGHT$(C$,1)
80 H=VAL(H$):IF H$>"9" THEN H=ASC(H
$)-55
85 L=VAL(L$):IF L$>"9" THEN L=ASC(L
$)-55
90 BY=H*16+L:PRINT#8,CHR$(BY);
95 NEXT:GOTO 10
100 REM COLOROUT 64 LM
101 DATA 01080D080A009E283230*363329
0000004C5311FE*C6DEDEECC0FE007E6666
102 DATA 7EE6E6E600FCCCCCFEC6*C6FE00
FECECEC0C0C6FE*00FCE6E6E6E6E6FC00FE
103 DATA C000FEF0F0FE00FEE0E0*FE6060
6000FECEC0C0DE*C6FE00C6C6C6FEE6E6E6
104 DATA 00181818383838380000C*0C0C0E
CECEFE00C6CCD8*FEE6E6E6006060600E0E0
105 DATA E0FE00C6EEFED6C6C6C6*00E6F6
CECEE6E6E600FE*C6C6CECECEFE00FEC6C6
106 DATA FEE0E0E000FEC6C6C6C6*F6FE18
FCE6E6FCCECECE*00FEC6C0FE06E6FE00FE
107 DATA 383838383838000C6C6C6*CECECE
FE00C6C6C6C6C6*6C3800C6C6C6D6FEEEC6
108 DATA 00C6C66C386CC6C600C6*C66C38
38383800FE7C18*3060C6FE007C60606060
109 DATA 607C00400C060301800C06*007C0C
0C0C0C0C7C0030*183C66000000000000000
110 DATA 000000FEFE000000000*000000
383838380003938*006666660000000000006C
111 DATA FE6C6CFE6C000030FC00*FC2CFC
300060660C1830*660600F8D8D876DCDCF6
112 DATA 0018181800000000000C*183030
30180C0030180C*0C0C1870000006C38FE38
113 DATA 6C00000018187E181800*000000
00003838183800*00007E7E000000000000
114 DATA 003838380000060C1830*60C000
FEC6C6CECECEFE*003878181878C3C00FE
115 DATA 0606FEC0C0FE00FC0C0C*FE0E0E
FE00C0C0C6C6FE*060600FEC0C0FE0606FE
116 DATA 00FCCCC0FEC6C6FE00FE*C60C18
383838007EE666*FEE6E6FE00FEC6C6FE0E
117 DATA 0E0E000038380038380000*000038
38003838183800C*19306030180C0000007E
118 DATA 007E0000006030180C18*306000
FEC61E18001818*0038109 2FE9210380000
119 DATA 00FE06FEC6FE00E0E0FE*C6C6C6
FE000000FEC6C0*C0FE000E0EFEC6C6C6FE
120 DATA 000000FEC6FEC0FE003C*30307C
303030000000FE*CECEFE06FEE0E0FEC6C6
121 DATA C6C6003800181638383B*000600
060606E6FE00C0*C0C6CCFEC6C600381818
122 DATA 1818183C000000C6EEFE*D6C600
0000FEC6C6C6C6*000000FECEE6E6FE0000
123 DATA 00FEC6C6FEE0E00000FE*C6C6FE
0E0E0000FEE6E0*E0E0000000FEC0FE06FE
```

COMPUTER E DIDATTICA

ESPERIENZE DIDATTICHE A CONFRONTO

L'AMLETO TECNODIDATTICO

«Scusi professore» mi fa un collega durante l'intervallo, «ci toccherà anche imparare a programmare?». Consapevole di quanto poco i miei colleghi s'interessino di computer e didattica, una simile domanda che, senza mezzi termini, va dritta dritta al nocciolo della questione, mi lascia senza parole. Al di là, infatti, delle considerazioni sui problemi psicologici del primo approccio all'elaboratore da parte degli insegnanti, il vero timore, l'autentica inespressa preoccupazione dei docenti è proprio questa: è obbligatorio imparare a programmare?

Dal volto del mio interlocutore traspare chiaramente il timore che io risponda con un netto "sì" che non ammette discussioni. Intuisco che, se lo facessi, in qualche modo lo ferirei. E per di più direi una bugia.

Partire col piede giusto

Quando gli insegnanti sentono parlare d'introduzione dell'informatica nella scuola, nella loro mente scattano due associazioni fondamentalmente errate: A) informatica = computer; B) computer a scuola = insegnante esperto di programmazione. Queste associazioni sembrano essere la causa scatenante di tutte le difficoltà, le paure e, di conseguenza, gli scetticismi, i pregiudizi, i rifiuti. Ed è pure sbagliato cadere nell'eccesso opposto: riporre cioè una fiducia esagerata nelle potenzialità della macchina e considerarla come la soluzione a tutti i problemi che quotidianamente un insegnante incontra facendo il proprio mestiere. Di questo passo, si finirebbe per abbracciare un ideale tecnocentrico foriero di un deludente nulla educativo. Non è il computer che "fa" scuola, ma è il docente che, con le sue cognizioni e la sua esperienza, intuisce e sfrutta le potenzialità didattiche della macchina. Bisogna sempre ricordare che prima, durante e dopo il computer c'è (e presumibilmente continuerà a esserci) il suo creatore, ossia l'uomo.

Quindi un atteggiamento equilibrato è quello di sgombrare la mente e studiare con serenità e occhio critico i modi più proficui in cui l'elaboratore può aiutare la didattica, senza mitizzarlo né scartarlo a priori. Conoscere la funzione, il ruolo, l'utilità che il mezzo informatico può avere nell'ambito di ciascuna materia dovrebbe far parte della formazione didattica di base di ogni insegnante e non solo di coloro che, per volontà propria o per dovere, si sono preoccupati di aggiornarsi partecipando al Piano Nazionale o frequentando corsi organizzati da vari enti (corsi, val la pena di sottolineare, spesso a pagamento e non riconosciuti dal Ministero).

Questa rubrica si basa su testimonianze di alunni ed insegnanti, che si interessano alla didattica del computer. I contributi editoriali sono grandemente apprezzati. Inviate eventuali materiali (articoli, foto, disegni, descrizioni di esperienze...) a:

COMMODORE GAZETTE
Comodore e didattica
Via Monte Napoleone, 9
20121 Milano

Saper scegliere: un questionario

Da quanto detto si può facilmente dedurre la risposta alla domanda iniziale: no, in linea di massima non è necessario che un insegnante conosca la programmazione per utilizzare didatticamente il computer in aula. Naturalmente è ovvio che la deve conoscere il docente titolare della materia in un I.T.I.S. da cui escono periti informatici. Ma è più importante avere altre abilità, altre capacità che non quella di saper scrivere software didattico.

Quali abilità? Quali capacità? Innanzi tutto quella di saper individuare e fissare criteri che permettano di scegliere il software (didattico e non)

più consono alle proprie esigenze nel marasma di proposte da parte di case editrici, "courseware house" e grandi industrie di computer. Vediamo un esempio di possibili criteri di scelta, schematizzandoli in un questionario di dieci domande.

Si constaterà che i criteri sono tesi a verificare e accertare: **(A)** la compatibilità tra software offerto e hardware disponibile a scuola, tenendo presente che con "hardware" non necessariamente s'intende l'elaboratore: alcuni pacchetti software, infatti, potrebbero essere stati concepiti per funzionare sia col computer sia col (video)registratore sia, magari, col lettore di videodischi (domanda 1); **(B)** il grado di complementarità tra il software offerto e la programmazione educativo-didattica (annuale e futura, in senso lato), gli approcci e i metodi seguiti dall'insegnante (domande 2, 3 e 9); **(C)** l'accettabilità del software da parte dell'alunno (domande 4 e 5); **(D)** la coerenza del software in sé, sia sotto il profilo della realizzazione tecnica sia sotto quello più propriamente didattico, riservando, a questo riguardo, una particolare attenzione al complesso di significati dei messaggi audiovisivi e testuali (domande 6, 7, 8); **(E)** la disponibilità di un'assistenza anche dopo l'adozione del software ed eventualmente la possibilità di sostituirlo nel caso d'inconvenienti che ne limitassero l'uso (scoperta tardiva di bug, danni ai dischi o altro) (domanda 10).

Questo, ovviamente, è solo un esempio di questionario e il docente può aggiungere, modificare o sostituire le domande a seconda delle proprie specifiche necessità e della materia insegnata.

Una delle maggiori difficoltà nella scelta e nella valutazione del software didattico è sovente dovuta all'impossibilità dell'insegnante di metterlo alla prova con calma. La pubblicità, gli opuscoli illustrativi, le parole del rappresentante non bastano, e talvolta la distribuzione è scadente: spesso le software house sono imprese medio-piccole (o anche privati) che si è costretti a contattare direttamente.

Altro software

Accanto ai programmi offerti da case editrici, industrie di elaboratori e software o courseware house, ve ne sono altri (etichettati, in genere, come educativi) della cui esistenza si viene a sapere soprattutto grazie al successo di una certa macchina: si veda, per esempio, la combinazione studiata tempo fa dalla Commodore per il "patriarca", il venerando C-64, che veniva offerto in abbinamento con ben due "kit scuola" alternativi, per non parlare dei programmi didattici che sfruttano le sorprendenti capacità dell'Amiga.

Come deve comportarsi l'insegnante in queste circostanze?

Benché virtualmente possa essere anch'esso messo al vaglio dei criteri ora illustrati, per formulare un giudizio di validità e utilità su questo software, l'insegnante deve far leva sostanzialmente sulla propria esperienza, professionalità e sensibilità di docente, prendendo e impiegando al meglio ciò che di buono esso offre.

Saper programmare? Ha i suoi vantaggi

Ebbene sì, anche se è innegabile che per un insegnante non sia indispensabile saper programmare, è d'altro canto fuor di dubbio che l'esercizio di quest'"arte", anche solo come passatempo, comporti dei vantaggi anche dal punto di vista didattico. Saper programmare significa, infatti, approfondire e rappresentare (e rappresentarsi) un problema o un concetto in maniera tale da poterlo logicamente, dettagliatamente e razionalmente "spiegare" al computer con le poche parole che conosce. Non è difficile intuire che una simile ginnastica mentale abitua il docente a una chiarezza d'analisi, sintesi, ed esposizione che si rivela utile anche nel più tradizionale confronto con il discente: la lezione.

A sostegno di questa tesi potremmo citare le parole di Richard Noss, coordinatore del progetto LOGO nell'ambito del Microelectronic Education Programme (MEP), il "programma" per l'introduzione dell'informatica nelle scuole britanniche: «La prima legge dell'informatica didattica è questa: il valore educativo di moltissimo software cosiddetto "educativo" è in gran parte riservato al programmatore, non all'utente» (Richard Noss, "Imparare a programmare in Logo", in Bill TAGG, Il software didattico per insegnanti e genitori, a cura di Giovanni Lariccia, Roma, Armando, 1988, pag. 108).

L'Amleto tecnodidattico

E allora? Programmare o non programmare? Il dilemma si fa amletico. La risposta rimane un no, anche se qualche esperienza di programmazione di base, come si è visto, non guasterebbe.

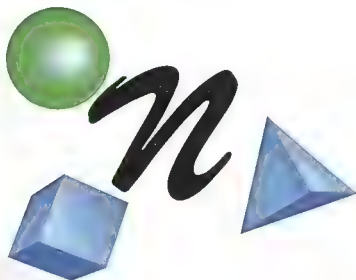
Stefano Franzato

Oltre al testo citato nell'articolo, si veda: Guida all'analisi del software didattico, *di Giorgio Olimpo e Michela Ott, De Agostini, Novara (in corso di stampa).* ■

ESEMPIO DI QUESTIONARIO PER LA SCELTA DEL SOFTWARE DIDATTICO

1) Il software propostomi può funzionare sulle apparecchiature tecnologiche in dotazione nella scuola?
2) Il software è sufficientemente integrabile con la mia programmazione didattica?
3) Il metodo didattico con cui il software affronta l'argomento si avvicina abbastanza al mio?
4) Il software è "amichevole" con l'utente cui è destinato (l'alunno)?
5) Il software presuppone che l'alunno possieda già qualche conoscenza sull'argomento trattato?
6) Il software è strutturato con coerenza?
7) Se il software fa uso di messaggi iconici e sonori, questi si integrano armonicamente col testo?
8) I manuali (d'obbligo in italiano) che accompagnano il software sono chiari, esaurienti (anche nei dettagli), arricchiti da esempi e scritti in uno stile scevro da tecnicismi inutili?
9) Se il software deve corredare un libro di testo, il rapporto tra i due è equilibrato?
10) Il venditore garantisce in ogni caso una buona assistenza, nonché eventualmente la rapida sostituzione del software in caso di difetti?

COMMODORE NEWS

NOVITÀ HARD E SOFTWARE DALL'ITALIA E DAL MONDO

ITALIA

NUOVA LINFA PER L'AMIGA

Tra gli importatori e distributori di pacchetti rivolti all'uso professionale o semi-professionale dell'Amiga, s'inserisce oggi la AP&S (Applied Peripherals & Software), un'azienda giovane e dinamica, il cui titolare, Massimiliano Pallavicini, entusiasta delle potenzialità dell'Amiga, ne vuole promuovere in particolare l'utilizzo professionale, importando e distribuendo sul mercato italiano dispositivi hardware e pacchetti software a prezzi estremamente competitivi.

Attualmente sono in fase di studio le modalità per l'istituzione di un servizio di supporto tecnico ai propri clienti, tramite il quale, al di fuori della regolare garanzia, possano essere risolti gli eventuali problemi che dovessero presentarsi nell'uso quotidiano dei prodotti distribuiti. Questo acquista particolare importanza per quanto riguarda il campo del software applicativo di uso professionale, per il quale è indispensabile disporre di strutture che offrano una reale assistenza post-vendita.

L'esordio di questa nuova azienda avviene con i titoli di alcune importanti software house (come New Horizons, Micro-Systems Software, Elan Design, SunRize Industries, Mindware, NewTek, Octree, Zuma Group). Per quanto riguarda gli accordi con la Micro-Systems, produttrice tra l'altro di *Scribble!*, *Excellence!*, *The Works!*, *On Line!* e *Organize!*, c'è da sottolineare la volontà dell'azienda di riservarsi l'esclusiva per la distribuzione in Italia. Sono infatti in corso promettenti trattative che hanno tra gli obiettivi la traduzione sia dei manuali sia del software.

Applied Peripherals & Software
Via Papa Giovanni, 37
33040 Corno di Rosazzo (UD)
(Tel. 0432/759264)
(Fax 0432/759264)

AMADEUS E MANNESMANN TALLY

Nato tre anni fa su iniziativa dell'Assessorato all'Educazione del Comune di Milano, il **Progetto Amadeus** per l'introduzione dell'informatica nella scuola si è attuato presso cinque scuole elementari, con viva soddisfazione degli insegnanti e con grande entusiasmo da parte dei bambini. Ora i corsi si estendono anche alle scuole medie: 76 ragazzi di prima e 84 di seconda potranno seguire gratuitamente, nel pomeriggio, corsi di programmazione in LOGO, di grafica e animazione, e di editoria elettronica. La Mannesmann Tally ha subito creduto in questo progetto e ha accettato, insieme ad altre aziende, di dare il suo contributo con la fornitura gratuita di stampanti per consentire l'allestimento dei 30 laboratori previsti dal progetto. Sono stati inviati alcuni tra i modelli più diffusi: la piccola e professionale MT 81, la collaudatissima MT 230 e la MT 905, una laser ormai nota per l'ottimo rapporto prezzo/prestazioni.

E a proposito della 9 aghi MT 81, ne segnaliamo il grosso successo testimoniato dagli ultimi dati di vendita: dal novembre 1988 alla fine del 1989, la Mannesmann Tally ha venduto oltre 22 mila esemplari della MT 81. Congratulazioni!

Mannesmann Tally Srl
Via Borsini, 6
20094 Corsico (MI)
(Tel. 02/4502850)

FIDO PER AMIGA

Segnaliamo la nascita di un nuovo BBS per Amiga 500 e PC IBM. Per il momento funziona in modo sperimentale, dalle 20,30 alle 8,30 di ogni giorno. Augurandogli un buon successo, ne pubblichiamo i dati di collegamento: telefono 0544/970157, baud da 300 a 2600, parity none, bit 8, stop 1.

IDEA!

La Software Copyright ha coniato un marchio con il quale distribuire parte del suo software: **Idea**. In

Software Gallery ne abbiamo recensito *Moonshadow* (il secondo titolo prodotto) ma vorremmo segnalare anche il primo: *Bomber Bob* per Amiga. Si tratta di uno shoot'em up con molti tocchi di umorismo. Grafica "stile fumetti", scroll verticale parallattico a due livelli, sei livelli di gioco, sezione bonus 3D, animazioni... e come se non bastasse, è stato firmato un contratto con Francesco Salvi per l'uso della sua canzone "Esatto!" all'interno del gioco.

S.C. Srl
Via Mazzini, 12
21020 Casciago (VA)
(Tel. 0332/222052)
(Fax 0332/228288)

SETTE NUOVE FUJITSU

Nel continuo sforzo di mantenere la leadership di fornitore di stampanti professionali ad alte prestazioni, la Fujitsu Italia annuncia sette nuovi modelli nell'area delle stampanti a matrice.

Le 9 aghi **DX2150** (L. 1.150.000) e **DX2250** (L. 1.350.000), rispettivamente da 80 e 136 colonne, vanno a sostituire i modelli DX2100 e DX2200. Le innovazioni più importanti riguardano il trattamento carta: nel pannello comandi è stato infatti aggiunto un tasto per l'inserimento automatico o l'espulsione automatica sia di fogli singoli che di moduli continui.

Sulla base dell'ormai collaudata meccanica della DX2400, sono stati prodotti due nuovi modelli per il mondo IBM: la prima si chiama **DX2400Tx** (L. 3.000.000), ed è destinata al collegamento Twinax con i sistemi della serie IBM System 3X e AS/400. La seconda si chiama **DX2400Cx** (L. 3.000.000) e può collegarsi ai controller IBM 3174, 3274/3276, 8775, Local Channel Attach e 4700 Device Cluster, emulando le stampanti coassiali IBM 3262/3268, 3287 e 4214 Model 1.

Per gli utenti che richiedono una gestione carta automatizzata al massimo è stato creato il modello **DL4400P** (L. 2.680.000 e L. 3.000.000 nella versione a colori). Infine, la serie delle 24 aghi dedicate agli alti volumi di stampa viene arricchita dalla **DL4600** (L. 3.100.000 e L. 3.420.000 per la versione a colori).

Fujitsu Italia Spa
Via Melchiorre Gioia, 8
20124 Milano
(Tel. 02/6572741)

KICK OFF 2

Una rivoluzione nel gioco del calcio: esce *Kick Off 2*, una versione completamente rinnovata del celebre programma vincitore del premio "gioco dell'anno" in Inghilterra.

Centinaia di scelte per i giocatori, controlli quasi istintivi di dribbling, passaggi, tiri, colpi di testa, e possibilità anche di entrate in scivolata e sforbiciate. Ci sono nove tipi di calci d'angolo. Possibilità di effettuare sostituzioni e di pianificare la tattica di gioco. Si possono anche effettuare partite "di coppa" con i tempi supplementari, e immagazzinare fino a dieci "azioni da goal" in memoria per rivederle quando si preferisce.

È prevista la possibilità di preparare speciali dischi-dati per simulare i grandi eventi calcistici come la Coppa del Mondo, le coppe europee e così via. Ma per una vera recensione è meglio aspettare di poter caricare il programma nei nostri disk drive.

Leader distribuzione Srl
Via Mazzini, 15
21020 Casciago (VA)
(Tel. 0332/212255)

INTERNATIONAL 3D TENNIS

È in arrivo *International 3D tennis* (Sensible Software), una rivoluzionaria simulazione tennistica. L'uso innovativo della grafica vettoriale ha permesso la creazione di un tennis con una serie di caratteristiche mai raggiunte finora da programmi analoghi.

Nuovo sistema di animazione 3D; quattro livelli di abilità (amatore, semiprofessionale, professionale e asso); vista del campo da dieci diverse angolazioni (infinite nella versione Amiga); possibilità d'imparare ed eseguire il dritto e il rovescio; un reale controllo della direzione dei colpi; ottima colonna sonora; opzione per due giocatori; salvataggio della partita in ogni momento; 64 possibili avversari con diversi livelli di abilità; 72 tornei da giocare, esatte riproduzioni di quelli originali; quattro diversi terreni di gioco (erba, terra battuta, tappeto e cemento); fotografie digitalizzate (Amiga); voci digitalizzate degli arbitri di Wimbledon...

Lago snc
Via Napoleona, 16
22100 Como
(Tel. 031/300174)

ESTERO

ASDG NEWS

È disponibile LAN Rover (per A2000/3000), la nuova scheda Ethernet della ASDG. Mette a disposizione i modi thick e thin, ed è perfettamente conforme allo standard IEEE 802.3; la scheda è dotata di un driver software che rende l'hardware immediatamente utilizzabile e che garantisce una velocità decisamente superiore a quella di tutte le altre concorrenti.

La stessa società annuncia anche la disponibilità di un nuovo sistema di elaborazione delle immagini (a 8 e 24 bit) che fornisce all'Amiga la massima velocità di rendering attualmente disponibile. Oltre a una lunga lista di effetti speciali, il programma (che è stato battezzato *The Art Department*) è in grado di produrre separazioni di colori di qualità professionale virtualmente da ogni immagine (fino a un massimo di 24 bitplane, cioè 16.777.216 colori).

In tutt'altro settore, la ASDG annuncia una nuova versione di *CygnusEd Professional*, con possibilità di undo illimitato, e una ricerca/sostituzione "turbo", cento volte più veloce di quella disponibile nella prima versione del prodotto.

ASDG Inc.
925 Stewart Street
Madison, Winsconsin 53713, USA
(Tel. 001/608/2736585)

■

Classified

Software

Scambio programmi per Amiga 500. Eventualmente acquisto se a buon prezzo. Annuncio sempre valido. Assicuro risposta. Telefono: 0171/269659.

Offro i migliori programmi di totocalcio, enalotto, totip, lotto, per Commodore 64. Contattare senza impegno: massima serietà, esperienza, rapidità. Luigi Cirillo – Via Ghiaie, 20 – 38100 Trento – Tel. 0461/930500.

Per Amiga metto a disposizione tantissimi programmi di ogni genere. Nessuno scopo di lucro. Invio liste a tutti. Scrivere a: Piero Cannizzaro – Casella Postale 3019 – 50127 Firenze.

Scambio programmi per Amiga, offro ed esigo massima serietà. Posseggo circa 1000 titoli di ogni genere. Rispondo a tutti. Inviare le vostre liste a: Vito Positano – Via Brunnelleschi, 2F – 70019 Triggiano (BA).

Vendo in blocco 50 dischetti con programmi vari a L. 100.000. Scrivere o telefonare a: Graziano Pavone – Via B. Croce, 6 – 65016 Montesilvano (PE) – Tel. 085/4451530.

Scambio programmi e manuali di ogni genere per Amiga. Escluso ogni scopo di lucro. Inviare liste a: Massimo Giagnoni – Via Selva, 3 – 50061 Compiobbi (FI).

Scambio, vendita, acquisto programmi per Commodore 128 (modi 128 e CP/M). Telefonare ai numeri: 011/4150675 – 011/6490116.

Cambio programmi, informazioni, manuali per Amiga. Telefonare al numero: 099/661022 dal venerdì alla domenica e chiedere di Francesco. Annuncio sempre valido.

Scambio programmi per C-128, sia in modo 128 che in CP/M. Programmi che sfruttano in pieno l'espansione 1750 e le 80 colonne. Annuncio sempre valido. Inviare liste a: Guido Baralla – C.P. 20 – 55040 Ripa (LU) – Tel. 0584/760693.

Cerco programmi per C-64 e tutto ciò che lo riguarda (informazioni di ogni genere). Rispondo a tutti. Grazie. Michele Franceschi – Via G. Caboto, 7 – 31059 Zero Branco (TV).

Vendo GEOS 128 2.0 nuovissimo con 3 dischi e 2 voluminosi manuali originali, confezione Lago a L. 80.000 - L. 110.000 aggiungendo un mouse nuovo Commodore. Telefonare a: Daniele – 051/982077 Bologna.

Hardware

Vendo Amiga 2000 doppio drive interno, monitor 1081, stampante MPS 1000, programmi grafica, giochi, WP e database anche originali a L. 2.500.000 (L. 2.000.000 senza stampante). Telefonare dopo le ore 20 al numero 089/794796.

Vendo Amiga 500 + mouse + 20 programmi tra i migliori a L. 700.000. Luca Masini – Tel. 0331/593871 dopo le 20.

Vendo scheda controller MD A2090A Commodore per Amiga 2000, hard disk Toshiba A2094A Commodore da 40M/28 ms, scheda A2058 con 2M espandibili a 8. Materiale originale, garantito Commodore, imballato, al 60% del listino. Luigi Callegari – Via De Gasperi, 47 – 21040 Sumirago (VA) – Tel. 0331/909183 (ore serali).

Cedo hardware per C-64/C-128/Amiga a prezzo interessante. Arrivi novità: Power Replay, la fantastica cartuccia per Amiga, Track-Drive, Syncro Express. Espansioni, drive, stampanti, dischetti Sony. Telefonare al numero: 080/686059 (ore serali).

Vendo espansione da 512K interna per Amiga 500, nuova in garanzia. Richiesta L. 160.000 (in regalo molto software). Scambio programmi Amiga con programmi in MS-DOS. Massima serietà. Gianni Cottogni – Via Strambino, 23 – 10010 Carrone (TO) – Tel. 0125/712311 (ore 18/21).

Vendo C-128D + mon. fosfori verdi + reg. cassette mod. C2N a L. 800.000. Regalo circa 30 cassette + 50 riviste + 6 manuali per C-128D. Vendo MC68020-16 a L. 120.000, MC68881-16 a L. 150.000, la coppia a L. 220.000. Tel. 0161/486422 dopo ore 20. Chiedere di Fabrizio.

Hardware Amiga realizzo a prezzi bassi. F. Lombardi – Via Pini, 4 – Rozzano (MI) – Tel. 02/8257433.

Vendo espansione di memoria per Amiga 500 da 512K, causa passaggio a sistema superiore, a solo L. 130.000. Tel. 0376/631594. Ore serali.

Vendesi nuova espansione di memoria Amiga 500 da 512K per L. 160.000. Telefonare a Giorgio – 079/294643.

Vendo C-128 + 1541 a L. 350.000; Syntetic Sound per Amiga 1000 a L. 100.000; cerco int. Midi per Amiga 1000. Dario Carcich – Via Udine, 51 – 33030 Rivolto (UD).

Vendo Commodore 128 + stampante 803 + supporto stampante + drive 1571 + modem + joystick + registratore + kit pulizia drive + 70 dischetti + 150 cassette + portadischi + portacassette. Migliaia di programmi; grafica, scrittura, corsi, giochi, ecc., corredati di manuali, riviste, libri, a L. 1.150.000. Tel. 0773/876442.

CLASSIFIED DELLA COMMODORE GAZETTE È UN MODO ECONOMICO PER INFORMARE LA PIÙ VASTA UTENZA COMMODORE SUI VOSTRI PRODOTTI O SERVIZI. GLI ANNUNCI NON A SCOPO DI LUCRO, INVIATI DA PRIVATI, VENGONO PUBBLICATI GRATUITAMENTE (COMPILARE L'APPOSITA SCHEDA DI SERVIZIO LETTORI).

Quote: 15.000 lire per linea, minimo 4 linee. Aggiungere 5.000 lire per ogni parola in grassetto o 50.000 lire per l'intero annuncio in grassetto.
Condizioni: pagamento anticipato. Vengono accettati assegni e vaglia postali. Gli assegni devono essere intestati a: IHT GRUPPO EDITORIALE s.r.l.
Forma: gli annunci sono soggetti all'approvazione dell'editore e devono essere scritti a macchina o in modo molto chiaro. Una linea equivale a 40 lettere, spazi tra le parole compresi. Pregasi sottolineare le parole che si intendono scrivere in grassetto.
Informazioni generali: gli inserzionisti devono sempre specificare nome e indirizzo completo. Gli annunci appariranno nel primo numero disponibile dopo il ricevimento.

Inviare il materiale a:
IHT GRUPPO EDITORIALE
UFFICI PUBBLICITARI
VIA MONTE NAPOLEONE, 9
20121 MILANO

Attenzione: Commodore Gazette non si assume responsabilità in caso di reclami di qualunque natura da parte degli inserzionisti e/o dei lettori. Nessuna responsabilità è altresì accettata per errori e/o omissioni di qualsiasi tipo.

Varie

L'Amiga, con proposte ecc., anche scambi con programmi, giochi e tutto ciò che lo riguarda; valido anche per il C-64 e lo Spectrum 48K. Scrivere o telefonare a: Renzo Reggio – Via Roma, 57 – 00032 Carpineto Romano (RM) – Tel. 06/9797134.

Vendo ***Commodore Gazette*** dal marzo 1988 al gennaio 1990, totale 11 fascicoli a L. 50.000. Solo in blocco. *Enigma* dal settembre 1988 al febbraio 1990, tredici fascicoli a L. 50.000. Maurizio Marotta – Via Aversano, 4 – 84100 Salerno.

Cerco software e hardware per Amiga a prezzi d'occasione. Tel. 039/879211 orari lavoro.

Help me!, involontariamente ho formattato

il disco di sistema della mia cartuccia MK V, chi ne fosse in possesso è pregato di contattarmi: Gianluca – C.P. 37 – 33082 Azzano Decimo (PN) – Cerco inoltre *Laserterm* per CBM 64 e adattatore 6499.

Scambio programmi per Amiga, vendo arretrati di *Commodore Gazette, Microcomputer* e *Amiga World* a metà prezzo. Marco Sivori – Via Barchetta, 18/9 – 16162 Bolzaneto (GE).

Scambiamo programmi, manuali e informazioni varie per Amiga e IBM XT/AT, scriveteci o telefonateci, rispondiamo a tutti (tranne che ai venditori). Tel. 015/882305 (Luca), 015/881703 (Doriano) – Software Computer Club – P.O. Box 39 – 13060 Valdengo (VC).

Desiderei contattare utenti Amiga che siano interessati alla computermusic. Inoltre scambio programmi e videogames per Amiga 500/2000. Annuncio sempre valido. No lucro. Telefonare al numero 099/699916, chiedendo di Renato.

Sono alla ricerca dei numeri arretrati di *Commodore Gazette* (solo se in buono stato) dal 1° numero al numero 1/88. Accetto solo offerte modiche. Lorenzo Martucci – Via Calvi, 6 – 74016 Massafra (TA).

Amici 64/128, facciamo vivere queste splendide macchine! Disponibili tutti i programmi con manuale. Assistenza soft e massima competenza, serietà, rapidità. Rivolgetevi con fiducia a: Giuliano Cinci – Pian dei Mantellini, 44 – 53100 Siena.

Cerco utenti di C-64 e C-128 per scambio consigli, trucchi e programmi. Max. serietà. Preferibilmente Modena e comuni limitrofi. Scrivere a: Enrico Pinelli – Via Poli, 9 – 41043 Formigine (MO).

Per Amiga cerco: manuali, public domain, trucchi di programmazione, integrati RAM da 4 x 256 Kilobit e utility di ogni genere. Sergio De Gregorio – Via Colle S. Pietro, 42 – 03100 Frosinone.

Commodore Club

È nato da una esigenza di sfruttare al meglio il vostro Amiga, il "Liguria User Club". Cerchiamo nuovi soci, sia per scambio software, che per eventuale adesione. Telefonateci o scriveteci adesso. Paolo Cladi – Via Betti, 5 San Giacomo 3/1 – 16035 Rapallo (GE) – Tel. 0185/51320.

Amiga Club cerca nuovi soci: scambia programmi per Amiga, disponibili ultime novità. Massima serietà. Tel. 0185/51320.

Amiga Club Napoli. Distribuzione gratuita software di pubblico dominio e shareware. Vasta disponibilità di manualistica tecnica. Per informazioni: Massimiliano Afiero – C.so Italia, isolato 11 – 80021 Afragola (NA) – Tel. 081/5527940.

Commodore Club Roma cerca soci con il computer C-64, Amiga 500 e 2000. Novità, software, programmi ecc. Ci sarà tutto. Iscrivetevi al miglior club del mondo. Conviene. Telefonate a Christian 06/6092709 o Daniel 06/6090913 Roma.

Amiga World Users è lieta di dare inizio ad una fantastica organizzazione finalizzata allo scambio di programmi, idee, esperienze e bibliografie. Biblioteca programmi illimitata con arrivi bisettimanali. Ultime novità dai migliori gruppi. Contattare il numero 0825/71710 chiedendo di Daniele. ■

INDICE DEGLI INSERZIONISTI

Direzione vendite spazi pubblicitari:

IHT Gruppo Editoriale – Commodore Gazette
Agente Pubblicitario: Aldo Pagano Pagano
Via Monte Napoleone, 9 – 20121 Milano
Tel. 02/794181 - 799492 - 792612 - 794122
Telex 334261 IHT I – Telefax 02/784021

Questo indice è da considerarsi come un servizio addizionale. L'Editore non si assume alcuna responsabilità per eventuali errori e/o omissioni.

Indirizzare eventuali lamentele riguardanti gli inserzionisti a:

Commodore Gazette
Uffici Pubblicitari
Relazioni Inserzionisti
Via Monte Napoleone, 9 – 20121 Milano

Nessuna responsabilità viene altresì assunta dalla Commodore Gazette per eventuali problemi di qualsiasi natura con gli inserzionisti. La responsabilità di quanto pubblicato negli spazi pubblicitari è esclusivamente del committente.

COME DIGITARE I LISTATI DI COMMODORE GAZETTE

I listati per C-64/128 contengono una particolare simbologia. Tutti i caratteri grafici e quelli di controllo sono stati tradotti in combinazioni di tasti facilmente comprensibili. Sono le istruzioni tra parentesi graffe. Per esempio, {SHFT L} indica che si deve tener premuto il tasto shift e premere una volta il tasto L. Ovviamente, non bisogna digitare le parentesi; quello che apparirà sullo schermo saranno simboli grafici. Altri esempi:

{20 SPAZI} premere la barra spaziatrice 20 volte.
{SHFT CLR} tenere premuto il tasto shift e premere una volta il tasto clr-home.
{2 CRSR ↓} premere cursore-giù due volte.
{CTRL 1} tenere premuto il tasto control e premere il tasto 1.
{COMD T} tenere premuto il tasto col logo Commodore e premere T.
{CRSR ←} premere cursore-sinistra una sola volta.
{SHFT A} tenere premuto il tasto shift e premere il tasto A.

Gli altri tasti che non danno origine a caratteri particolari (come ↑, £, @) sono invece presentati normalmente.

Manoscritti: le collaborazioni dei lettori – manoscritti, disegni e/o fotografie – sono benvenute e verranno valutate in vista di una possibile pubblicazione. Commodore Gazette non si assume comunque responsabilità per perdite o danni al materiale. Si prega di allegare una busta affrancata e indirizzata per ogni articolo. Il pagamento per materiale non richiesto viene effettuato solo in seguito all'accettazione da parte della redazione. I contributi editoriali (di qualunque forma) non si restituiscono. Tutta la corrispondenza editoriale, richieste di annunci, problemi di sottoscrizione abbonamenti, di diffusione e con gli inserzionisti, deve essere indirizzata a: Commodore Gazette – Uffici Editoriali – Via Monte Napoleone, 9 – 20121 Milano. **Commodore Gazette** è un periodico indipendente non connesso in alcun modo con la Commodore Business Machines e con tutte le sue sussidiarie e affiliate, compresa la Commodore Italiana S.p.A. Commodore Gazette viene pubblicata dalla IHT Gruppo Editoriale, Via Monte Napoleone 9, 20121 Milano. Nessuna parte di questa pubblicazione può essere in alcun modo riprodotta senza il permesso scritto dell'editore. La redazione si adopera per fornire la massima accuratezza negli articoli e nei listati pubblicati. Commodore Gazette non si assume responsabilità per eventuali danni dovuti a errori od omissioni.

IL PROSSIMO NUMERO SARÀ IN EDICOLA I PRIMI DI SETTEMBRE

SERVIZIO LETTORI

Questa scheda è valida fino al 31 agosto 1990

A. Come giudica questo numero di Commodore Gazette?

- ☐ 1. Ottimo
- ☐ 2. Molto buono
- ☐ 3. Buono
- ☐ 4. Discreto
- ☐ 5. Sufficiente
- ☐ 6. Mediocre
- ☐ 7. Insufficiente

B. Quale(i) articolo(i) di questo numero ha apprezzato maggiormente? ______

C. Quale(i) articolo(i) di questo numero giudica peggiore(i)? ______

D. Quali argomenti dovrebbero essere trattati nei prossimi numeri di Commodore Gazette? ______

E. Con quale aggettivo descriverebbe Commodore Gazette? ______

F. Quante persone leggono la sua copia di Commodore Gazette?

- ☐ 1. Uno
- ☐ 2. Due
- ☐ 3. Tre
- ☐ 4. Quattro o più

G. Ha dei suggerimenti? ______

H. Quale(i) computer utilizza?

- ☐ 1. C-64
- ☐ 2. C-128
- ☐ 3. C-128D
- ☐ 4. Amiga 500
- ☐ 5. Amiga 1000
- ☐ 6. Amiga 2000
- ☐ 7. Altro (specificare) ______

I. Quale(i) computer intende acquistare nel futuro?

- ☐ 1. C-64
- ☐ 2. C-128D
- ☐ 3. Amiga 500
- ☐ 4. Amiga 2000
- ☐ 5. Altro (specificare) ______

L. È un acquirente dei libri della IHT? Se sì come li giudica? ______

M. Quali altre riviste (sia d'informatica che non) legge abitualmente? ______

N. Indichi quali sono i suoi maggiori interessi

- ☐ 1. Videoregistrazione
- ☐ 2. Hi-Fi
- ☐ 3. Strumenti musicali
- ☐ 4. Fotografia
- ☐ 5. Automobili
- ☐ 6. Sport
- ☐ 7. Viaggi

O. Quali periferiche intende acquistare nei prossimi sei mesi? ______

P. Quanto intende spendere in software e hardware nei prossimi sei mesi? ______

Nome ______
Cognome ______
Indirizzo ______
Città ______
Prov. ______ **C.a.p.** ______ **Età** ______
Professione ______

COMMODORE GAZETTE **luglio 1990**

SCHEDA ORDINAZIONE LIBRI

Con il presente tagliando desidero ordinare i(l) seguenti(e) libri(o):

Collana Informatica

- ☐ L'Amiga (Michael Boom) .... L. 60.000
- ☐ Il Manuale dell'AmigaDOS (Commodore-Amiga) .... L. 60.000
- ☐ Programmare l'Amiga Vol. I (Eugene P. Mortimore) .... L. 80.000
- ☐ Programmare l'Amiga Vol. II (Eugene P. Mortimore) .... L. 70.000
- ☐ Guida ufficiale alla programmazione di GEOS (Berkeley Softworks) .... L. 64.000
- ☐ Flight Simulator Co-pilot (Charles Gulick) .... L. 30.000
- ☐ Volare con Flight Simulator (Charles Gulick) .... L. 45.000

Collana Cinema

- ☐ Le mille luci di Hollywood (David Chell) .... L. 42.000

Collana Tempus

- ☐ Inventori del nostro tempo (Kenneth A. Brown) .... L. 42.000
- ☐ Computer in guerra: funzioneranno? (David Bellin e Gary Chapman) .... L. 39.900
- ☐ La sfida della crescita (G. Ray Funkhouser e Robert R. Rothberg) .... L. 39.900
- ☐ La Macchina e la Mente (George Johnson) .... L. 42.000
- ☐ I Creatori del Domani (Grant Fjermedal) .... L. 39.900

Pagherò in contrassegno al postino la somma di L. .... + spese postali (L. 6.000 per volume)

Nome e cognome ______
Indirizzo ______
Città ______
Prov. ______ **C.a.p.** ______ **Tel.** ______
Firma ______

COMMODORE GAZETTE **luglio 1990**

☐ Desidero inserire gratuitamente un mio annuncio nella rubrica CLASSIFIED (solo per i privati e per gli annunci non a scopo di lucro).

☐ Desidero inserire un mio annuncio nella rubrica CLASSIFIED. Allego assegno bancario o circolare o postale oppure fotocopia della ricevuta del vaglia postale per un totale di L. ______________________ . Il mio codice fiscale o partita IVA (per le aziende) è il seguente:______________________.

Attenzione: perché un annuncio venga accettato è necessario che sia stato compilato anche il questionario presente sull'altro lato di questo tagliando. Non si accettano fotocopie.

TESTO:______________________

____

____

____

Inserire all'interno di una busta affrancata e spedire a:

Commodore Gazette
Servizio Lettori
Via Monte Napoleone, 9
20121 Milano

- - - - - ✂ -

Inserire all'interno di una busta affrancata e spedire a:

IHT Gruppo Editoriale
Divisione Libri
Via Monte Napoleone, 9
20121 Milano

GEOS
NON HA PIÙ SEGRETI

Sei un utente di GEOS e ne sei entusiasta? Desideri scoprirne ogni dettaglio e impararne il funzionamento? Vuoi programmare in questo sistema operativo ad ambiente grafico? Allora questo volume della IHT ti è indispensabile. Scritta dagli stessi creatori di GEOS, questa guida svela tutti i segreti del sistema operativo e permette di creare stupende applicazioni per il C-64 e il C-128 dotate di interfaccia utente grafica, menu, icone, finestre, box di dialogo, fonti proporzionali, processi in multitasking, RAM disk, gestione dei file in overlay, controllo del mouse, driver di input e di stampa, e tante altre caratteristiche che fanno di GEOS un ambiente di lavoro e sviluppo davvero professionale.

592 pagine, L. 64.000

I libri IHT sono disponibili nelle migliori librerie e computer shop. Per ordini diretti servirsi dell'apposito modulo pubblicato a pagina 95

IHT Gruppo Editoriale - Via Monte Napoleone, 9 - 20121 Milano - Tel. 02/794181-794122 - Fax 784021 - Telex 334261 IHT I

Distribuzione: RCS Rizzoli Libri - Via Scarsellini, 17 - 20161 Milano - Tel. 02/64068508